オリエンタルモーター株式会社
ステッピングモーターユニットαSTEP
高効率 AR シリーズ FLEX
位置決め機能内蔵タイプ(AC/DC 電源入力)

サンプル画面説明書

三菱電機株式会社

サンプルのご利用について

サンプル用の画面データ、取扱説明書などのファイルは、以下の各項に同意の上でご利用いただくものとします。

(1) 当社製品をご使用中またはご使用検討中のお客様がご利用の対象となります。

(2) 当社が提供するファイルの知的財産権は、当社に帰属するものとします。

(3) 当社が提供するファイルは、改竄、転載、譲渡、販売を禁止します。
但し、内容の一部または全てをお客様作成の機器やシステム内の当社製品上でご利用いただく場合は、その限りではありません。また、当社製品をご利用いただいたお客様作成の仕様書、設計書、組み込み製品の取扱説明書などへの転載、複製、引用、レイアウトの変更についてもその限りではありません。

(4) 当社が提供するファイルやそのファイルから抽出されるデータを利用することによって生じた如何なる損害も当社は補償をいたしません。お客様の責任においてご利用ください。

(5) 当社が提供するファイルに利用条件などが添付されている場合は、その条件にも従ってください。

(6) 予告なしに当社が提供するファイルの削除や内容の変更を行うことがあります。

(7) 当社が提供するファイルのご使用に際しては、対応するマニュアルおよびマニュアルで紹介している関連マニュアルをよくお読みいただくと共に、安全に対して十分に注意を払って正しい取扱いをしてください。

目次

改訂履歴

サンプル画面説明書

改訂日付	管理番号*	改訂内容
2014/8	BCN-P5999-0415	初版

＊ 管理番号は、右下に記載しています。

プロジェクトデータ

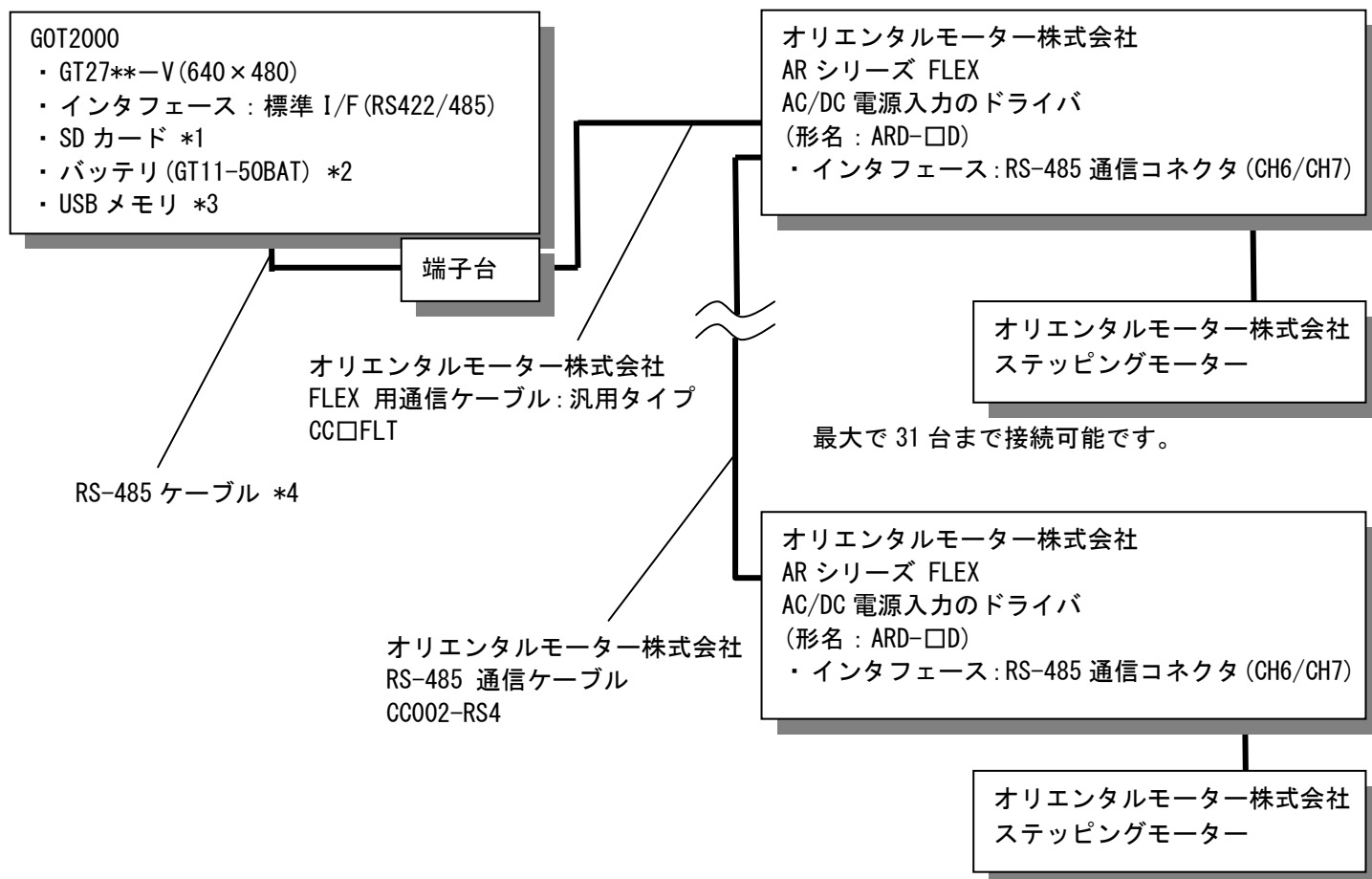
改訂日付	プロジェクトデータ	GT Designer3*	改訂内容
2014/8	ORIENTAL_AR-MODBUS_V_Ver1_J.GTX	1.117X	初版

＊ プロジェクトデータ作成時に使用した作画ソフトウェアのバージョンです。記載したバージョンと同等、またはそれ以降のバージョンの作画ソフトウェアを使用してください。

1. 概要

GOT2000 とオリエンタルモーター株式会社 AR シリーズ FLEX のドライバ(ARD-□D)をシリアル(RS-485)で接続し、オリエンタルモーター株式会社 ステッピングモーターの現在値や設定値のモニタ、変更を行うサンプル画面の説明書です。

2. システム構成

2.1 システム構成

*1：SD カードは、ドキュメント表示機能・レシピ機能で使用しています。
*2：バッテリは、時計データの停電保持に使用しています。(バッテリは GOT 本体に標準装備しています。)
*3：USB メモリは、レシピ機能で使用しています。
*4：接続方法の詳細については、「GOT2000シリーズ接続マニュアル(マイコン・MODBUS・周辺機器接続編)」を参照してください。

3. GOT について

3.1 自動で選択されるシステムアプリケーション

種類	システムアプリケーションの名称		
基本機能	基本システムアプリケーション		
	標準フォント	日本語	
通信ドライバ	MODBUS/RTU		
拡張機能	標準フォント		中国語(簡体)
	アウトラインフォント	ゴシック	英数かな
			日本語漢字
			中国(簡体)漢字
	レシピ操作		
	ドキュメント表示		

3.2 作画ソフトウェアの接続機器の設定

項目	設定値	備考
ボーレート(BPS)	115200	
データ長	8 bit	
ストップビット	1 bit	
パリティ	偶数	
リトライ回数(回)	2	
通信タイムアウト時間(秒)	3	
自局アドレス	1	モニタするコントローラの軸番号を設定します。
送信ディレイ時間(ms)	8	8ms 以上で設定してください。
32bit 格納順序	HL 順	HL 順で設定してください。
ファンクションコード[0F]	使用しない	
ファンクションコード[10]	使用する	
コイル読出し点数(点)	2000	
入力リレー読出し点数(点)	2000	
保持レジスタ読出し点数(点)	16	最大点数は 16 点です。
入力レジスタ読出し点数(点)	125	
コイル書込み点数(点)	800	
保持レジスタ書込み点数(点)	16	最大点数は 16 点です。

3.3 作画ソフトウェアのオーバーラップウィンドウ設定

ベース画面の切り換え時にウィンドウ画面を閉じるために、[画面切り換え/ウィンドウ]のオーバーラップウィンドウの[詳細設定]で[ベース画面の切り換えと同時にウィンドウを閉じる]を有効にしています。

4. ドライバについて

4.1 AC 電源ドライバの通信設定

弊社で動作確認した際の設定値は下記となります。

(1)　パラメータの設定

項目	設定値	備考
通信ストップビット	1 bit	
通信パリティ	偶数	

(2)　ドライバのディップスイッチ、ロータリスイッチの設定

項目	設定値	備考
号機設定スイッチ(ID)	ID=1	接続するドライバごとに値を変更
通信速度設定スイッチ(SW2)	SW2=4	115200bps
機能設定スイッチ(SW4)	No.2=ON	ON：Modbus プロトコルを選択
終端抵抗設定スイッチ(TERM.)	No.1、No.2=ON	最終端のドライバのみ終端抵抗をON に設定する

4.2 DC 電源ドライバの通信設定

弊社で動作確認した際の設定値は下記となります。

(1)　パラメータの設定

項目	設定値	備考
通信ストップビット	1 bit	
通信パリティ	偶数	

(2)　ドライバのディップスイッチ、ロータリスイッチの設定

項目	設定値	備考
号機設定スイッチ(SW1)	ID=1	接続するドライバごとに値を変更
通信速度設定スイッチ(SW2)	SW2=4	115200bps
機能設定スイッチ(SW3 No.2)	No.2=ON	ON：Modbus プロトコルを選択
終端抵抗設定スイッチ(SW3 No.4)	No.4=ON	最終端のドライバのみ終端抵抗をON に設定する

4.3 サンプルの適用対象ドライバ

サンプルは、以下の製造年月日もしくはドライババージョンのドライバに対応しています。

(1)　製造年月日

2014 年 1 月以降

*製造年月日は、ドライバの銘板に記載しています。

(2)　ドライババージョン

AC 電源入力ドライバ：Ver.2.00 以降

DC 電源入力ドライバ：Ver.2.01 以降

*ドライババージョンは、データ設定ソフト MEXE02 のステータス,I/O モニタで確認できます。

5. 画面仕様

5.1 表示言語

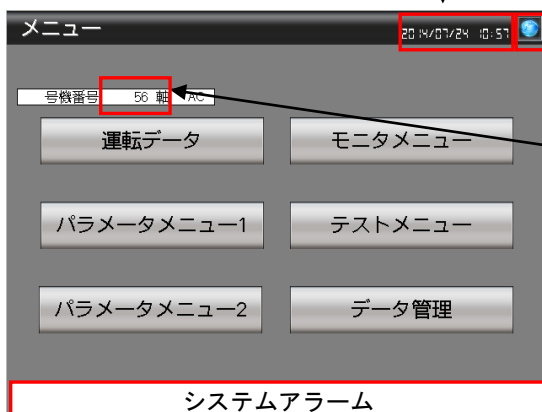
画面上に表示する文字列は、日本語・英語・中国語(簡体)の 3 言語で切り換え表示できます。各言語の文字列は、コメントグループ No.497～500 の列 No.1～3 に下記のように登録しています。言語切り換えデバイスに列 No.を格納すると列 No.に対応した言語を表示します。

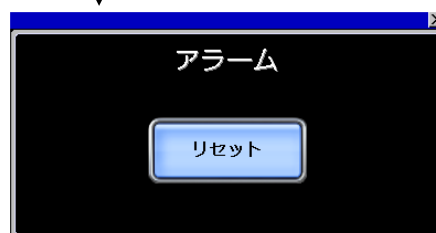
列 No.	言語
1	日本語
2	英語
3	中国語(簡体)

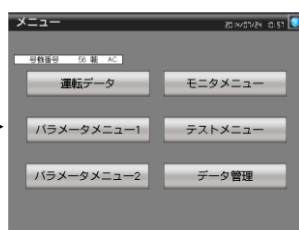
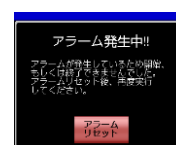
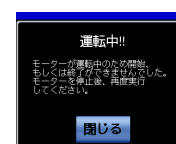
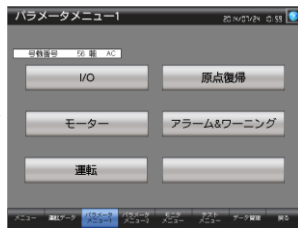
5.2 画面一覧・遷移

5.2.1 画面一覧・遷移(共通)

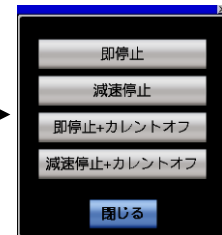
ウィンドウ画面 W-30003：時計設定

ウィンドウ画面 W-30002：言語設定

ベース画面：全ベース画面

ウィンドウ画面 W-30004：軸切り換え(B-30001以外で表示)

ウィンドウ画面 W-30001：アラームリセット

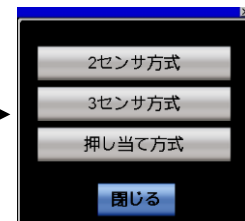
5.2.2 画面一覧・遷移(個別)

次項へ

前項より

ベース画面 B-30003：
パラメータメニュー1

ベース画面 B-31004：
パラメータ I/O

ウィンドウ画面 W-32002：
STOP 入力停止方法

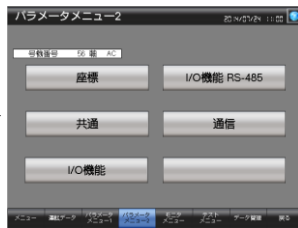
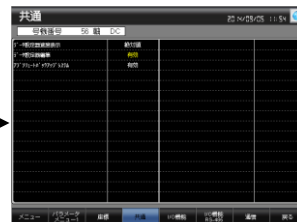
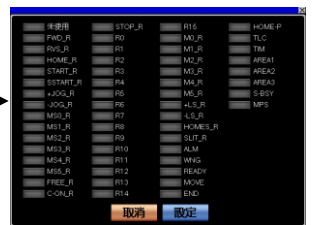
ベース画面 B-31005：
パラメータ モーター

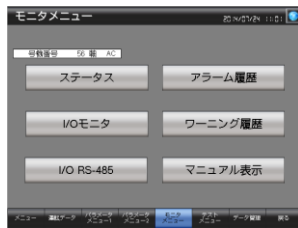
ベース画面 B-31006：
パラメータ 運転

ベース画面 B-31007：
パラメータ 原点復帰

ウィンドウ画面 W-32003：
原点復帰方法

ベース画面 B-31008：
パラメータ アラーム&ワーニング

次項へ

前項より

ベース画面 B-30004：
パラメータメニュー2
ベース画面 B-31010：
パラメータ 座標
ベース画面 B-31011：
パラメータ 共通
ウィンドウ画面 W-32004：
IN 入力機能選択
ベース画面 B-31012：
パラメータ I/O 機能
ウィンドウ画面 W-32006：
OUT/NET-OUT 出力機能選択
ベース画面 B-31013：
パラメータ I/O 機能 RS-485
ウィンドウ画面 W-32005：
NET-IN 入力機能選択
ベース画面 B-31014：
パラメータ 通信
ウィンドウ画面 W-32006：
OUT/NET-OUT 出力機能選択

次項へ

前項より

ベース画面 B-30005：
モニタメニュー

ベース画面 B-31015：
モニタ ステータス

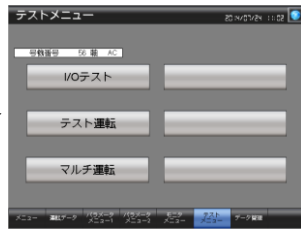
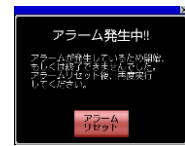
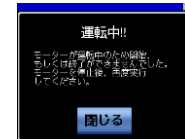
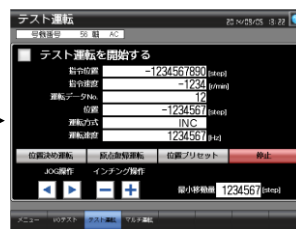
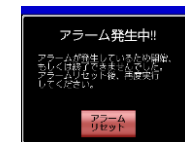
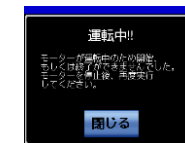
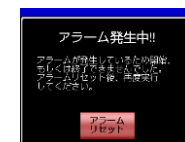
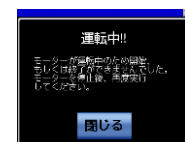
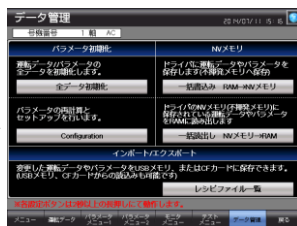
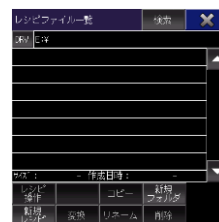
ベース画面 B-31016：
モニタ I/O モニタ

ベース画面 B-31017：
モニタ I/O RS-485

ベース画面 B-31018：
モニタ アラーム履歴

ベース画面 B-31019：
モニタ ワーニング履歴

次項へ

ベース画面 B-30500～30502：
マニュアル表示-言語 1～3

前項より
ベース画面 B-31020：
テスト I/O テスト
ウィンドウ画面 W-32007：
アラーム発生中確認
ウィンドウ画面 W-32008：
モーター運転中確認
ベース画面 B-31021：
テスト 運転
ウィンドウ画面 W-32007：
アラーム発生中確認
ウィンドウ画面 W-32008：
モーター運転中確認
ベース画面 B-31022～31029：
テスト マルチ運転
ウィンドウ画面 W-32007：
アラーム発生中確認
ウィンドウ画面 W-32008：
モーター運転中確認
ベース画面 B-31030：
システム データ管理
システム画面
レシピファイル一覧ウィンドウ

5.3 画面説明

5.3.1 AC/DC 選択画面(B-30001)

概要

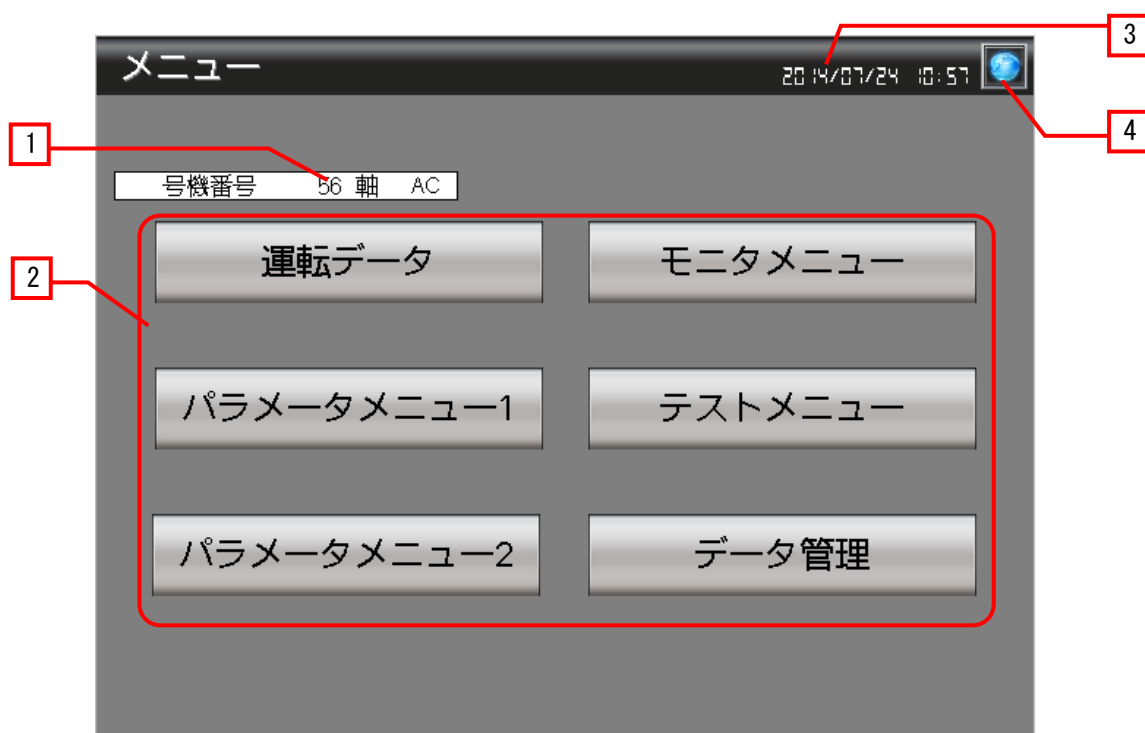
AC/DC 電源を指定します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、号機番号を変更できます。
2. モニタするドライバの電源タイプを指定します。
3. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
4. 言語設定ウィンドウを表示します。

備考

- 複数台のドライバをモニタする場合は、接続機器設定の自局アドレスで設定した号機番号のドライバが必ず存在するようにしてください。サンプルでは自局アドレスに「1」を設定しています。自局アドレス設定の詳細については、「GOT2000 シリーズ接続マニュアル(マイコン・MODBUS・周辺機器接続編) GT Works3 Version1 対応」を参照してください。
- GOT 起動時に、プロジェクトスクリプトにて号機番号に「1」を設定しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。
- システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.2　メニュー(B-30002)

概要

メニュー画面です。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 各画面に切り換えます。
3. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
4. 言語設定ウィンドウを表示します。

備考

・ システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.3 運転データ(B-31002)

概要

ドライバの運転データを表示・変更します。また、ステッピングモーターを操作します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 運転データを表示します。タッチした運転データ No.の運転データを運転データ入力ウィンドウから設定できます。
3. モーターの指令位置を表示します。
4. ティーチング運転を開始するにチェックを入れると、ティーチング運転を開始します。

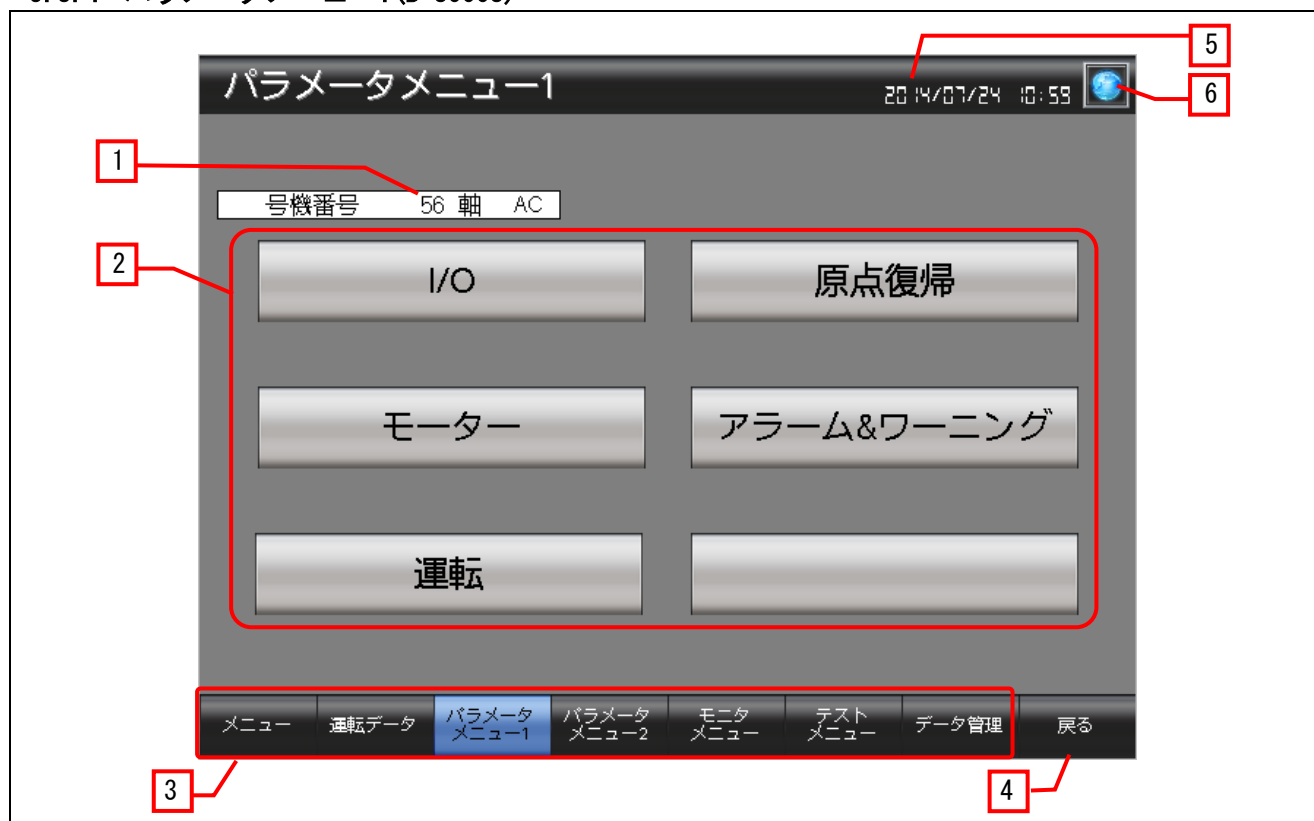
最小移動量	：モーターが動く最小移動量の設定ができます。
運転データ No.	：運転データ No.を選択します。
◀▶	：スイッチを押している間、正転・逆転の連続運転をします。 運転データ No.で選択した No.の運転速度、加速、減速になります。
−+	：モーターの位置を調整します。 最小移動量で設定した移動量分だけモーターが動きます。
停止	：運転中のモーターを停止します。
位置決め運転	：運転データ No.で選択した No.で位置決め運転を実行します。
原点復帰運転	：原点復帰運転を開始します。
位置確定	：モーターがいる位置を運転データ No.で選択した No.の位置に反映します。また、運転方式をアブソリュート(ABS)に変更します。
位置プリセット	：指令位置を任意の値にプリセットします。プリセットする値は、パラメータ 座標「プリセット位置」の項目で変更できます。

5. 運転データをスクロールします。

▲	：8 件分、上へスクロールします。
▼	：8 件分、下へスクロールします。

6. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
7. 前回表示していた画面に切り換えます。
8. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
9. 言語設定ウィンドウを表示します。

備考
・ティーチング運転の開始や終了をするたびに、一瞬モーターの励磁が切れますので、上下軸などでお使いの場合はご注意をお願いします。また、一瞬モーターの励磁が切れて、再励磁するため位置ずれが発生します。位置精度を求める場合は、ティーチング運転を終了した直後に、原点復帰運転を実行することをお勧めします。 ・ティーチング運転で、加速、減速が選択した運転データの設定に変わるのは、加減速選択パラメータの設定が独立になっている時のみとなります。 ・ティーチング運転中は他画面に切り換えることや号機番号の変更はできません。 ・ティーチング運転の実行は画面スクリプトを使用しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。 ・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.4 パラメータメニュー1(B-30003)

概要

パラメータメニュー1 画面です。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 各画面に切り換えます。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.5 パラメータ I/O(B-31004)

概要

I/O に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. I/O に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
・LS 接点設定、HOMES 接点設定、SLIT 接点設定を変更した場合は、必ず「Configuration」コマンドを実行してください。「Configuration」コマンドを実行しないと、変更した設定は、反映されません。「Configuration」コマンドは、データ管理画面で、実行できます。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.6 パラメータ モーター(B-31005)

概要

モーターに関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. モーターに関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
・フィルタ選択、制御モード、スムースドライブを変更した場合は、必ず「Configuration」コマンドを実行してください。「Configuration」コマンドを実行しないと、変更した設定は、反映されません。「Configuration」コマンドは、データ管理画面で、実行できます。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.7 パラメータ 運転(B-31006)

概要

運転に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 運転に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
・加減速単位、自動復帰動作を変更した場合は、必ず「Configuration」コマンドを実行してください。
「Configuration」コマンドを実行しないと、変更した設定は、反映されません。
「Configuration」コマンドは、データ管理画面で、実行できます。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.8 パラメータ 原点復帰(B-31007)

概要

原点復帰に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 原点復帰に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.9 パラメータ アラーム&ワーニング(B-31008)

概要

アラームとワーニングに関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. アラームとワーニングに関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
・原点復帰未完了アラームを変更した場合は、必ず「Configuration」コマンドを実行してください。
「Configuration」コマンドを実行しないと、変更した設定は、反映されません。
「Configuration」コマンドは、データ管理画面で、実行できます。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.10 パラメータメニュー2(B-30004)

概要

パラメータメニュー2 画面です。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 各画面に切り換えます。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.11 パラメータ 座標(B-31010)

概要

座標に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 座標に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

- パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
- 電子ギア A、電子ギア B、モーター回転方向、ラウンド設定、ラウンド設定範囲を変更した場合は、必ず「Configuration」コマンドを実行してください。「Configuration」コマンドを実行しないと、変更した設定は、反映されません。「Configuration」コマンドは、データ管理画面で、実行できます。
- システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.12 パラメータ 共通(B-31011)

概要

共通に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 共通に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
・アブソリュートバックアップシステムを変更した場合は、必ず「Configuration」コマンドを実行してください。「Configuration」コマンドを実行しないと、変更した設定は、反映されません。
「Configuration」コマンドは、データ管理画面で、実行できます。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.13 パラメータ I/O 機能(B-31012)

概要

I/O 機能に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. I/O 機能に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。

・I/O 機能のパラメータを変更した場合は、必ず「Configuration」コマンドを実行してください。
「Configuration」コマンドを実行しないと、変更した信号の機能は、反映されません。
「Configuration」コマンドは、データ管理画面で、実行できます。

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.14 パラメータ I/O 機能 RS-485(B-31013)

概要

I/O 機能 RS-485 に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. I/O 機能 RS-485 に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。

・I/O 機能 RS-485 のパラメータを変更した場合は、必ず「Configuration」コマンドを実行してください。
「Configuration」コマンドを実行しないと、変更した信号の機能は、反映されません。
「Configuration」コマンドは、データ管理画面で、実行できます。

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.15 パラメータ 通信(B-31014)

概要

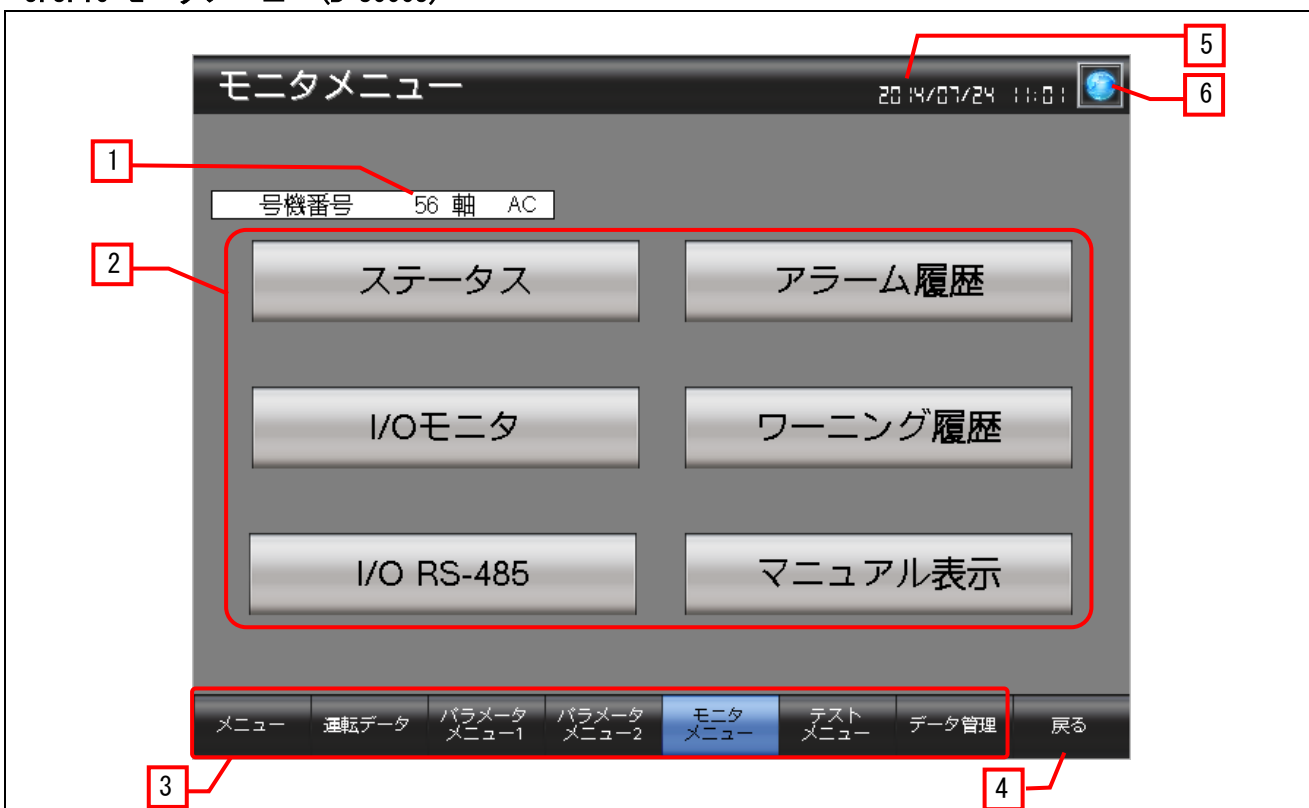
通信に関するパラメータを表示、編集します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 通信に関するパラメータを表示、編集します。数値、文字が黄色の場合は、初期値をあらわしています。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・パラメータの設定範囲は、電源のタイプによって異なる箇所があります。詳細は、ドライバの取扱説明書を参照してください。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.16 モニタメニュー(B-30005)

概要

モニタメニュー画面です。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 各画面に切り換えます。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 前回表示していた画面に切り換えます。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.17 モニタ ステータス(B-31015)

概要

モーターのステータスをモニタします。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 各項目の状態を表示します。
3. 現在発生中のアラームとワーニングを表示します。
4. 絶対位置異常アラームをリセットします。リセット後、原点復帰運転などを実行し、原点を再設定してください。
5. 現在発生中のアラームをリセットします。
6. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
7. 前回表示していた画面に切り換えます。
8. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
9. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.18 モニタ I/O モニタ (B-31016)

概要

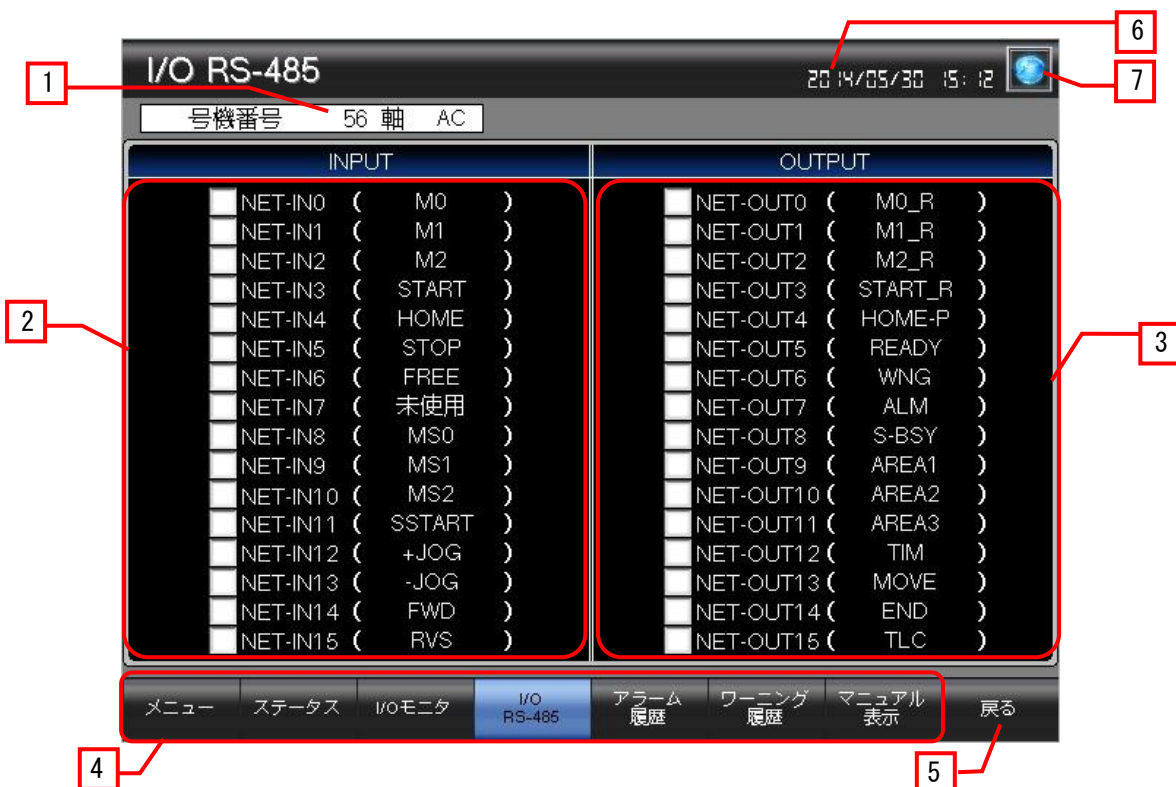
I/O をモニタします。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. INPUT の状態を表示します。
3. OUTPUT の状態を表示します。
4. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
5. 前回表示していた画面に切り換えます。
6. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
7. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.19 モニタ I/O RS-485(B-31017)

概要

ネットワーク通信用の I/O をモニタします。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. INPUT の状態を表示します。
3. OUTPUT の状態を表示します。
4. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
5. 前回表示していた画面に切り換えます。
6. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
7. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.20 モニタ アラーム履歴(B-31018)

概要

アラーム履歴を表示します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 現在発生中のアラームとワーニングを表示します。
3. アラームの履歴を表示します。
4. 絶対位置異常アラームをリセットします。リセット後、原点復帰運転などを実行し、原点を再設定してください。
5. 現在発生中のアラームをリセットします。
6. アラーム履歴をクリアします。
7. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
8. 前回表示していた画面に切り換えます。
9. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
10. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.21 モニタ ワーニング履歴(B-31019)

概要

ワーニング履歴を表示します。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 現在発生中のアラームとワーニングを表示します。
3. ワーニングの履歴を表示します。
4. 絶対位置異常アラームをリセットします。リセット後、原点復帰運転などを実行し、原点を再設定してください。
5. 現在発生中のアラームをリセットします。
6. ワーニング履歴をクリアします。
7. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
8. 前回表示していた画面に切り換えます。
9. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
10. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.22 マニュアル表示-言語 1(B-30500)、言語 2(B-30501)、言語 3(B-30502)

概要

表示中の言語に対応したマニュアルを表示します。

詳細

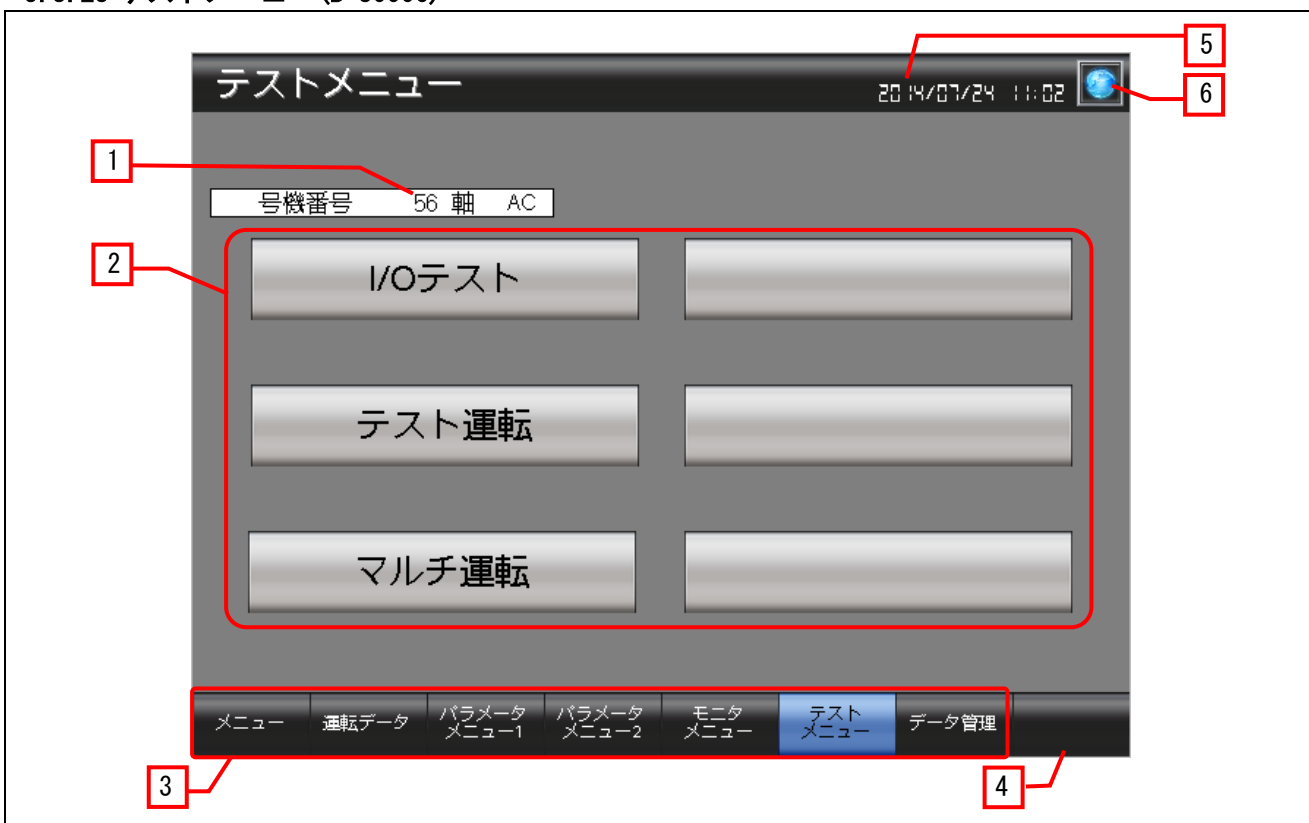
1. マニュアル表示-言語 1(B-30500)～言語 3(B-30502)は、それぞれドキュメント ID 201～203 のドキュメントを表示します。画面初回表示時は 1 ページ目を表示します。ドキュメント上をタッチした状態で 8 方向にフリックするとドキュメントを 8 方向にスクロール表示します。ドキュメントの端が表示されている状態でフリックすると、ページを切り換えます。ピンチアウト・ピンチインすると、大・中・小の 3 段階で、ドキュメントが切り換わります。
2. 表示しているドキュメントを操作します。
   - ：表示しているドキュメントを拡大/縮小します。
   - ：表示しているドキュメントを左右にスクロールします。
   - ：表示しているドキュメントを上下にスクロールします。
3. 表示しているドキュメントのページを操作します。
   - P. 1 ：表示しているドキュメントのページ数を表示します。数値をタッチするとページ番号を変更できます。
   - ：表示しているドキュメントをページ送り/ページ戻しします。
4. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
5. 前回表示していた画面に切り換えます。
6. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
7. 言語設定ウィンドウを表示します。

備考

- マニュアルの言語は、マニュアル用に作成するドキュメントの言語となります。マニュアル以外のタイトルやスイッチ銘板の言語は、コメントグループ No.497～500 の列 No.1～3 に設定されているコメントの言語となります。ドキュメント(ドキュメント ID)とコメントグループ No.497～500 の列 No.の関係は下表となります。

ベース画面	ドキュメント ID	列 No.
マニュアル表示-言語 1(B-30500)	201	1
マニュアル表示-言語 2(B-30501)	202	2
マニュアル表示-言語 3(B-30502)	203	3

- GOT 起動時に、プロジェクトスクリプトにてドキュメントページ No.に「1」を設定しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。
- マニュアル表示用のドキュメントデータは、お客様で作成してください。詳細については、「6.マニュアル表示について」を参照してください。
- システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.23 テストメニュー(B-30006)

概要

テストメニュー画面です。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. 各画面に切り換えます。
3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 未使用の画面切り換えスイッチです。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.24 テスト I/O テスト(B-31020)

概要

入力の確認と出力のテストをします。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. チェックボックスをタッチすると I/O テストを開始します。
3. INPUT の状態を表示します。
4. OUTPUT の状態を表示します。スイッチをタッチすることで OUT0～OUT5 出力の状態を確認できます。
5. 発生中のアラームをリセットします。
6. 現在発生中のアラームを表示します。
7. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
8. 未使用の画面切り換えスイッチです。
9. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
10. 言語設定ウィンドウを表示します。

備考

・I/O テストの開始や終了をするたびに、一瞬モーターの励磁が切れますので、上下軸などでお使いの場合はご注意をお願いします。また、一瞬モーターの励磁が切れて、再励磁するため位置ずれが発生します。位置精度を求める場合は、I/O テストを終了した後に、原点復帰運転を実行することをお勧めします。
・I/O テスト中は他画面に切り換えることや号機番号の変更はできません。
・I/O テストの実行は画面スクリプトを使用しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.25 テスト 運転(B-31021)

概要

モーターのテスト運転をします。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. チェックボックスをタッチするとテスト運転を開始します。

指令位置	：指令位置を表示します。
指令速度	：指令速度を表示します。
運転データ No.	：運転データ No. を選択します。
位置	：位置を表示します。
運転方式	：運転方式を表示します。
運転速度	：運転速度を表示します。
原点復帰運転	：原点復帰運転を開始します。
位置決め運転	：運転データ No. で選択した No. で位置決め運転を実行します。
位置プリセット	：指令位置を任意の値にプリセットします。プリセットする値は、パラメータ 座標「プリセット位置」の項目で変更できます。
停止	：運転中のモーターを停止します。
◀ ▶	：スイッチを押している間、正転・逆転の連続運転をします。 運転データ No. で選択した No. の運転速度、加速、減速になります。
− ＋	：モーターの位置を調整します。 最小移動量で設定した移動量分だけモーターが動きます。
最小移動量	：モーターが動く最小移動量の設定ができます。

3. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
4. 未使用の画面切り換えスイッチです。
5. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
6. 言語設定ウィンドウを表示します。

備考

・テスト運転の開始や終了をするたびに、一瞬モーターの励磁が切れますので、上下軸などでお使いの場合はご注意をお願いします。また、一瞬モーターの励磁が切れて、再励磁するため位置ずれが発生します。位置精度を求める場合は、テスト運転を終了した直後に、原点復帰運転を実行することをお勧めします。
・テスト運転で、加速、減速が選択した運転データの設定に変わるのは、加減速選択パラメータの設定が独立になっている時のみとなります。
・テスト運転中は他画面に切り換えることや号機番号の変更はできません。
・テスト運転の実行は画面スクリプトを使用しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.26 テスト マルチ運転(B-31022～31029)

概要

複数軸のモーターのテスト運転をします。

詳細

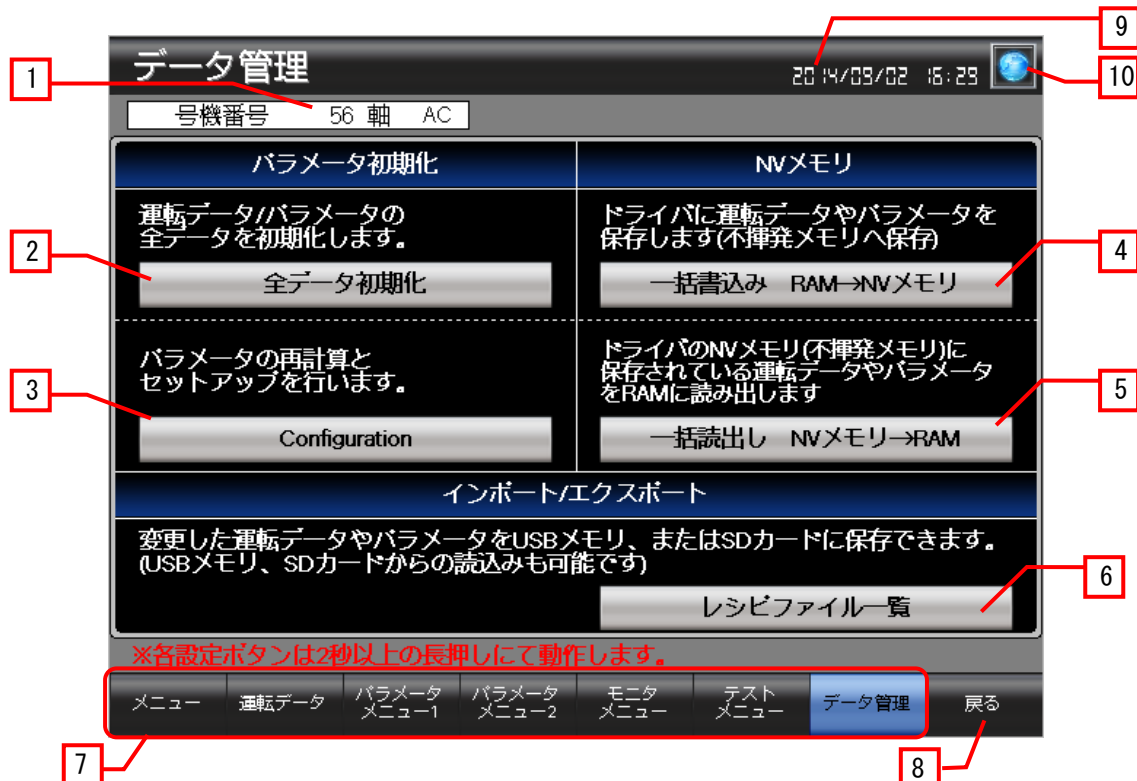
1. モーターのテスト運転を実行する軸を選択します。
2. 各軸スイッチをタッチすることで、その軸のテスト運転ができます。
3. 発生中のアラームを表示します。
4. 各軸のテスト運転を実行します。

データ No.	: 運転データ No. を選択します。
◀▶	: スイッチを押している間、正転・逆転の連続運転をします。 運転データ No. で選択した No. の運転速度、加速、減速になります。
−+	: モーターの位置を調整します。 運転パラメータの「JOG 移動量」で設定した移動量分だけ動作します。
位置決め運転	: 運転データ No. で選択した No. で位置決め運転を実行します。
位置プリセット	: 指令位置を任意の値にプリセットします。プリセットする値は、パラメータ 座標「プリセット位置」の項目で変更できます。
停止	: 運転中のモーターを停止します。
原点復帰運転	: 原点復帰運転を開始します。
アラームリセット	: 発生しているアラームをリセットします。

5. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
6. 未使用の画面切り換えスイッチです。
7. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
8. 言語設定ウィンドウを表示します。

備考

・テスト運転の開始・終了をするたびに、一瞬モーターの励磁が切れますので、上下軸などでお使いの場合はご注意をお願いします。また、一瞬モーターの励磁が切れて、再励磁するため位置ずれが発生します。位置精度を求める場合は、テスト運転を開始・終了した直後に、原点復帰運転を実行することをお勧めします。
・テスト運転で、加速、減速が選択した運転データの設定に変わるのは、加減速選択パラメータの設定が独立になっている時のみとなります。
・テスト運転中は他画面に切り換えることや号機番号の変更はできません。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。

5.3.27 システム データ管理(B-31030)

概要

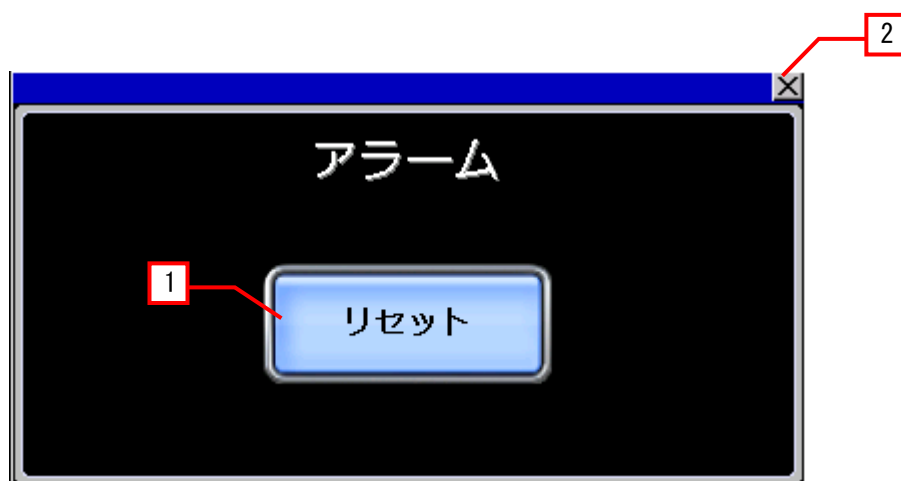
ドライバの初期化、Configuration、運転データやパラメータの保存、読出しをします。

詳細

1. モニタするドライバの号機番号を表示します。タッチすると、軸切り換えウィンドウを表示します。
2. ドライバの NV メモリに保存されている、運転データやパラメータを全て初期化します。
3. ソフトウェアリセットが実行されます。パラメータによってはソフトウェアリセットを実行しないと反映されないものもあります。
4. RAM に保存されている運転データやパラメータを NV メモリに保存します。
5. NV メモリに保存されている運転データやパラメータを RAM に読出します。
6. レシピファイル一覧ウィンドウを表示します。運転データやパラメータを USB メモリや SD カードに保存、読出しをします。
7. 各画面に切り換えます。青色のスイッチは、現在表示中画面のため画面は切り換わりません。
8. 前回表示していた画面に切り換えます。
9. 現在の日時を表示します。タッチすると、時計設定ウィンドウを表示します。
10. 言語設定ウィンドウを表示します。

備考

・全てのスイッチ操作は 2 秒以上長押しで実行します。
・「Configuration」コマンドを実行すると、一瞬モーターの励磁が切れますので、上下軸などでお使いの場合はご注意をお願いします。また、一瞬モーターの励磁が切れて、再励磁するため位置ずれが発生します。位置精度を求める場合は、Configuration」コマンドを操作した直後に、原点復帰運転を実行することをお勧めします。
・システムアラームが発生した場合、画面下部にアラームメッセージを表示します。メッセージの左端をタッチすると、表示位置が画面上部、画面中央、画面下部の順に切り換わります。それ以外をタッチすると、アラームリセットウィンドウが表示されます。
・運転データやパラメータの保存、読出しの詳細については、「8.運転データやパラメータの USB メモリ、SD カードへの保存、読出しについて」を参照してください。

5.3.28 アラームリセット(W-30001)

概要

システムアラームをリセットします。

詳細

1. システムアラームをリセットし、1秒後にウィンドウ画面を閉じます。
2. ウィンドウ画面を閉じます。

備考

5.3.29 言語設定(W-30002)

概要

GOT で表示する言語を選択します。

詳細

1. 言語を切り換え、ウィンドウ画面を閉じます。
2. ウィンドウ画面を閉じます。

備考

・表示言語にあわせてシステム言語も切り換える設定をしています。
・マニュアル表示-言語 1～言語 3 のいずれかのベース画面を表示中に言語設定ウィンドウで言語を切り換えた場合、選択した言語に対応したマニュアル表示画面に切り換わるように画面スクリプトを設定しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。

5.3.30 時計設定(W-30003)

概要

GOT の時計データを変更します。

詳細

1. 現在の日時を表示します。
2. 変更したい日時を▼▲スイッチで設定します。▼▲スイッチは、長押しすると連続で増減します。リセットスイッチは、秒をリセットします。
3. 設定した日時を GOT の時計データに反映し、1 秒後にウィンドウ画面を閉じます。
4. ウィンドウ画面を閉じます。

備考

・変更する日時の初期値は、ウィンドウ画面を表示した時の日時です。
・変更する日時の年・月・日・時・分・秒の数値表示にオブジェクトスクリプトを設定しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。

5.3.31 軸切り換え(W-30004)

概要

号機選択と電源タイプを切り換えます。

詳細

1. モニタする号機を選択します。
2. モニタする号機のドライバの電源タイプを指定します。
3. 設定を反映せずに、ウィンドウ画面を閉じます。
4. 設定を反映し、ウィンドウを閉じます。号機選択、電源タイプの選択がされてない場合は、スイッチをタッチしても動作しません。

備考

・ 号機選択、電源タイプの設定は画面スクリプトを使用しています。スクリプトの詳細については「5.6 スクリプト一覧」を参照してください

5.3.32 運転データ入力(W-32001)

概要

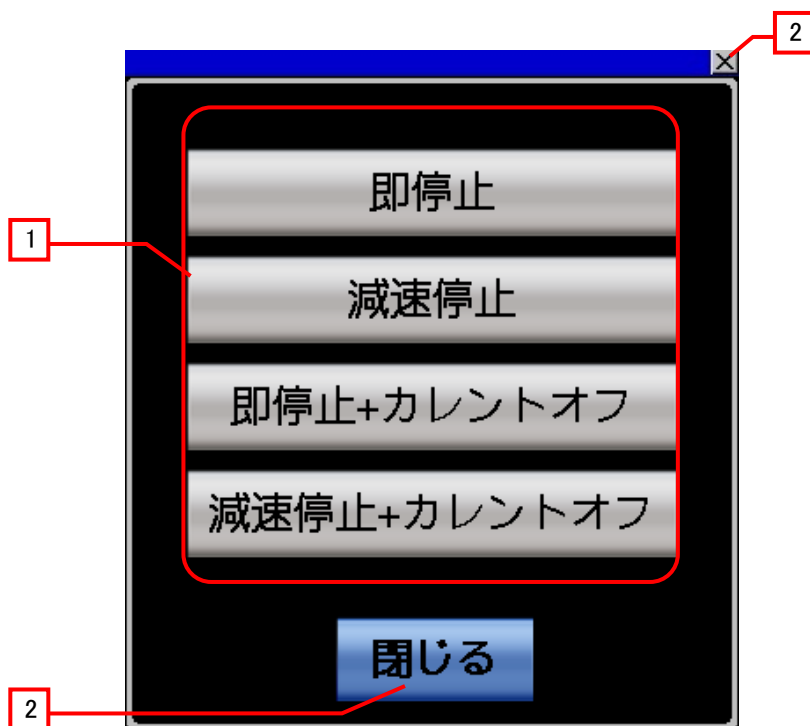
運転データを表示、編集します。

詳細

1. 運転データを表示、編集します。
2. 入力用テンキーです。
3. 編集したデータを反映せずに、ウィンドウ画面を閉じます。
4. 編集したデータを反映し、ウィンドウ画面を閉じます。

備考

・運転データの読出し、書込みは画面スクリプトを使用しています。スクリプトの詳細については「5.6 スクリプト一覧」を参照してください。

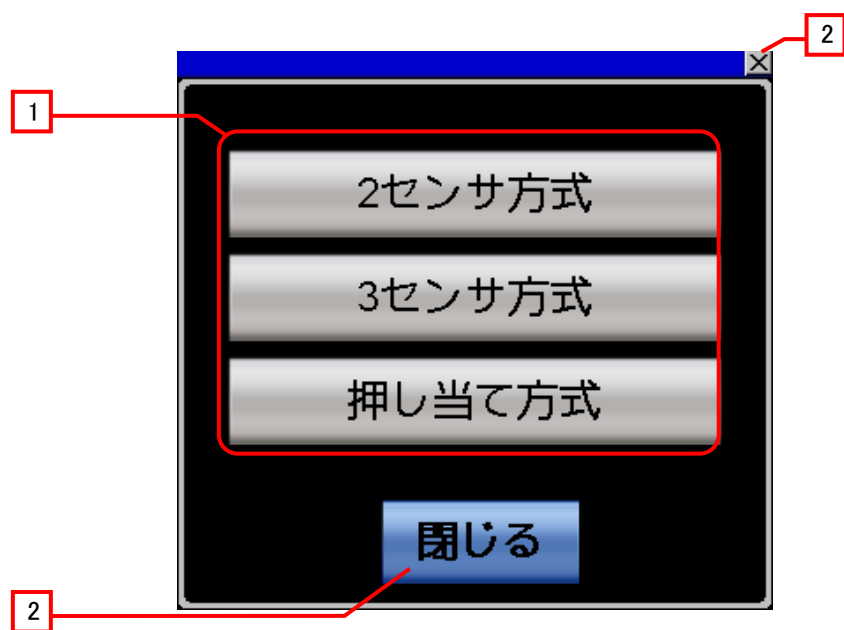
5.3.33 STOP 入力停止方法(W-32002)

概要

STOP 入力停止方法の設定内容を表示、設定します。

詳細

1. STOP 入力停止方法の設定内容を表示、設定します。
2. ウィンドウ画面を閉じます。

備考

5.3.34 原点復帰方法(W-32003)

概要

原点復帰方法の設定内容を表示、設定します。

詳細

1. 原点復帰方法の設定内容を表示、設定します。
2. ウィンドウ画面を閉じます。

備考

5.3.35 IN 入力機能選択(W-32004)

概要

IN 入力機能選択の設定内容を表示、設定します。

詳細

1. IN 入力機能選択の設定内容を表示、設定します。
2. 設定を反映せずに、ウィンドウ画面を閉じます。
3. 設定を反映し、ウィンドウ画面を閉じます。

備考

・設定内容の書込みは画面スクリプトを使用しています。スクリプトの詳細については「5.6 スクリプト一覧」を参照してください。

5.3.36 NET-IN 入力機能選択(W-32005)

概要

NET-IN 入力機能選択の設定内容を表示、設定します。

詳細

1. NET-IN 入力機能選択の設定内容を表示、設定します。
2. 設定を反映せずに、ウィンドウ画面を閉じます。
3. 設定を反映し、ウィンドウ画面を閉じます。

備考

・設定内容の書込みは画面スクリプトを使用しています。スクリプトの詳細については「5.6 スクリプト一覧」を参照してください。

5.3.37 OUT/NET-OUT 出力機能選択(W-32006)

概要

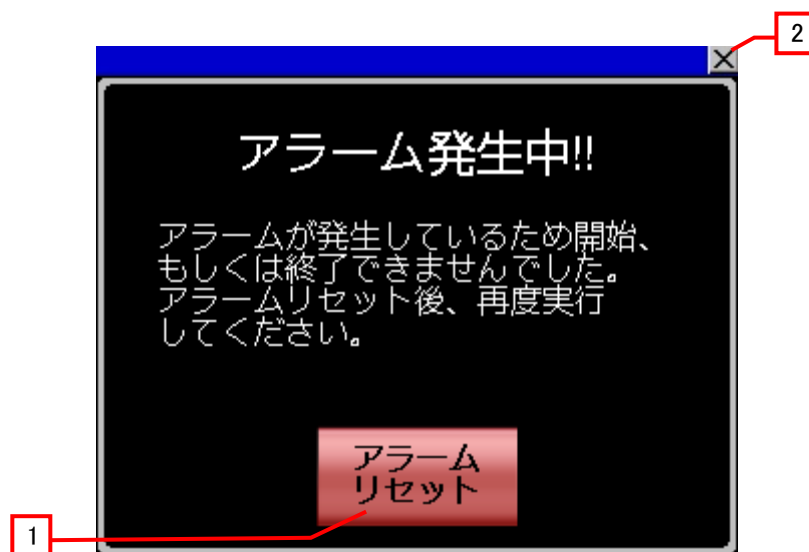
OUT/NET-OUT 出力機能選択の設定内容を表示、設定します。

詳細

1. OUT/NET-OUT 出力機能選択の設定内容を表示、設定します。
2. 設定を反映せずに、ウィンドウ画面を閉じます。
3. 設定を反映し、ウィンドウ画面を閉じます。

備考

・MPS については電源タイプが AC の場合のみ設定可能です。
・設定内容の書込みは画面スクリプトを使用しています。スクリプトの詳細については「5.6 スクリプト一覧」を参照してください。

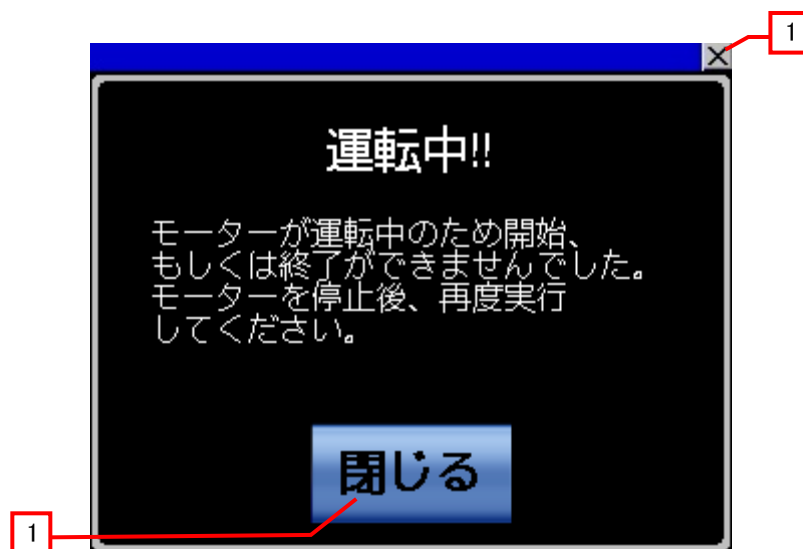
5.3.38 アラーム発生中確認(W-32007)

概要

アラーム発生中にティーチング、各テスト運転を開始、終了しようとした場合、通知します。

詳細

1. 発生中のアラームをリセットし、ウィンドウ画面を閉じます。
2. ウィンドウ画面を閉じます。

備考

5.3.39 モーター運転中確認(W-32008)

概要

モーター運転中にティーチング、各テスト運転を開始、終了しようとした場合、通知します。

詳細

1. ウィンドウ画面を閉じます。

備考

5.4 使用デバイス一覧

画面上のスイッチやランプなどに設定されている一部のデバイスは、スクリプトなどの共通設定にも設定されている場合があります。これらのデバイスを一括で変更する場合には[一括変更]の使用を推奨します。[一括変更]の詳細については、「GT Designer3 (GOT2000) ヘルプ」を参照してください。

5.4.1 接続機器のデバイス

タイプ	デバイス番号	用途
ビット	400126.b0	ドライバ入力指令[M0]
	400126.b1	ドライバ入力指令[M1]
	400126.b2	ドライバ入力指令[M2]
	400126.b3	ドライバ入力指令[START]
	400126.b4	ドライバ入力指令[HOME]
	400126.b5	ドライバ入力指令[STOP]
	400126.b6	ドライバ入力指令[FREE]
	400126.b7	ドライバ入力指令[未使用]
	400126.b8	ドライバ入力指令[MS0]
	400126.b9	ドライバ入力指令[MS1]
	400126.b10	ドライバ入力指令[MS2]
	400126.b11	ドライバ入力指令[SSTART]
	400126.b12	ドライバ入力指令[+JOG]
	400126.b13	ドライバ入力指令[-JOG]
	400126.b14	ドライバ入力指令[FWD]
	400126.b15	ドライバ入力指令[RVS]
	400128.b0	ドライバ出力指令[M0_R]
	400128.b1	ドライバ出力指令[M1_R]
	400128.b2	ドライバ出力指令[M2_R]
	400128.b3	ドライバ出力指令[START_R]
	400128.b4	ドライバ出力指令[HOME-P]
	400128.b5	ドライバ出力指令[READY]
	400128.b6	ドライバ出力指令[WNG]
	400128.b7	ドライバ出力指令[ALM]
	400128.b8	ドライバ出力指令[S-BSY]
	400128.b9	ドライバ出力指令[AREA1]
	400128.b10	ドライバ出力指令[AREA2]
	400128.b11	ドライバ出力指令[AREA3]
	400128.b12	ドライバ出力指令[TIM]
	400128.b13	ドライバ出力指令[MOVE]
	400128.b14	ドライバ出力指令[END]
	400128.b15	ドライバ出力指令[TLC]
	400213.b0	ダイレクトI/O、電磁ブレーキの状態[OUT0]
	400213.b1	ダイレクトI/O、電磁ブレーキの状態[OUT1]
	400213.b2	ダイレクトI/O、電磁ブレーキの状態[OUT2]
	400213.b3	ダイレクトI/O、電磁ブレーキの状態[OUT3]
	400213.b4	ダイレクトI/O、電磁ブレーキの状態[OUT4]
	400213.b5	ダイレクトI/O、電磁ブレーキの状態[OUT5]
	400214.b0	ダイレクトI/O、電磁ブレーキの状態[+LS]
	400214.b1	ダイレクトI/O、電磁ブレーキの状態[-LS]
	400214.b2	ダイレクトI/O、電磁ブレーキの状態[HOMES]
	400214.b3	ダイレクトI/O、電磁ブレーキの状態[SLIT]
	400214.b6	ダイレクトI/O、電磁ブレーキの状態[IN0]
	400214.b7	ダイレクトI/O、電磁ブレーキの状態[IN1]
	400214.b8	ダイレクトI/O、電磁ブレーキの状態[IN2]
	400214.b9	ダイレクトI/O、電磁ブレーキの状態[IN3]

タイプ	デバイス番号	用途
ビット	400214.b10	ダイレクトI/O、電磁ブレーキの状態[IN4]
	400214.b11	ダイレクトI/O、電磁ブレーキの状態[IN5]
	400214.b12	ダイレクトI/O、電磁ブレーキの状態[IN6]
	400214.b13	ダイレクトI/O、電磁ブレーキの状態[IN7]
	400386.b0	アラームのリセット
	400388.b0	絶対位置異常アラームのリセット
	400390.b0	アラーム履歴クリア
	400392.b0	ワーニング履歴クリア
	400396.b0	P-PRESET 実行
	400398.b0	Configuration 実行
	400400.b0	全データ初期化
	400402.b0	NV メモリー括読み出し
	400404.b0	NV メモリー括書き込み
	400516.b0	ハードウェアオーバートラベル
	400518.b0	オーバートラベル動作
	400538.b0	LS 接点設定
	400540.b0	HOMES 接点設定
	400542.b0	SLIT 接点設定
	400654.b0	加減速選択
	400656.b0	加減速単位
	400716.b0	原点復帰開始方向
	400718.b0	原点復帰 SLIT センサ検出
	400720.b0	原点復帰 TIM 信号検出
	400778.b0	原点復帰未完了アラーム
	400902.b0	モーター回転方向
	400904.b0	ソフトウェアオーバーラベル
	400912.b0	ラウンド設定
	400962.b0	データ設定器速度表示
	400964.b0	データ設定器編集
	400966.b0	アブソリュートバックアップシステム
	404110.b0	HOME-P 出力機能選択
	404130.b0	フィルタ選択
	404136.b0	制御モード
	404138.b0	スムースドライブ
	404162.b0	自動復帰動作
	404386.b0	IN0 入力接点設定
	404388.b0	IN1 入力接点設定
	404390.b0	IN2 入力接点設定
	404392.b0	IN3 入力接点設定
	404394.b0	IN4 入力接点設定
	404396.b0	IN5 入力接点設定
	404398.b0	IN6 入力接点設定
	404400.b0	IN7 入力接点設定
ワード	400126	ドライバ入力指令(下位)
	400129	現在のアラーム(上位)
	400130	現在のアラーム(下位)
	400132+2n (n=0～9)	アラーム履歴 1～10(上位)
	400152	現在のワーニング(下位)
	400154+2n (n=0～9)	ワーニング履歴 1～10(上位)
	400195	現在の選択データ No.(上位)
	400196	現在の選択データ No.(下位)
	400197	現在の運転データ No.(上位)
	400199	指令位置(上位)

タイプ	デバイス番号	用途
ワード	400201	指令速度(上位)
	400205	フィードバック位置(上位)
	400207	フォードバック速度(上位)
	400395	P-PRESET 実行(上位)
	400397	Configuration(上位)
	400513	STOP 入力停止方法(上位)
	400514	STOP 入力停止方法(下位)
	400515	ハードウェアオーバートラベル(上位)
	400517	オーバートラベル動作(上位)
	400519	位置決め完了出力幅(上位)
	400521	位置決め完了出力オフセット(上位)
	400523	AREA1 +方向位置(上位)
	400525	AREA1 -方向位置(上位)
	400527	AREA2 +方向位置(上位)
	400529	AREA2 -方向位置(上位)
	400531	AREA3 +方向位置(上位)
	400533	AREA3 -方向位置(上位)
	400535	MOVE 出力最小時間(上位)
	400537	LS 接点設定(上位)
	400539	HOMES 接点設定(上位)
	400541	SLIT 接点設定(上位)
	400577	RUN 電流(上位)
	400579	STOP 電流(上位)
	400581	位置ループゲイン(上位)
	400583	速度ループゲイン(上位)
	400585	速度ループ積分時定数(上位)
	400587	速度フィルタ(上位)
	400589	移動平均時間(上位)
	400641	共通加速(上位)
	400643	共通減速(上位)
	400645	起動速度(上位)
	400647	JOG 運転速度(上位)
	400649	JOG 加速度(上位)
	400651	JOG 起動速度(上位)
	400653	加減速選択(上位)
	400655	加減速単位(上位)
	400656	加減速単位(下位)
	400705	原点復帰方法(上位)
	400706	原点復帰方法(下位)
	400707	原点復帰運転速度(上位)
	400709	原点復帰加速度(上位)
	400711	原点復帰起動速度(上位)
	400713	原点復帰オフセット(上位)
	400715	原点復帰開始方向(上位)
	400717	原点復帰 SLIT センサ検出(上位)
	400719	原点復帰 TIM 信号検出(上位)
	400721	押し当て原点復帰運転電流(上位)
	400769	過負荷アラーム(上位)
	400771	カレントオン時位置偏差過大アラーム(上位)
	400777	原点復帰未完了アラーム(上位)
	400833	過熱ワーニング(上位)
	400835	過負荷ワーニング(上位)
	400837	過速度ワーニング(上位)

タイプ	デバイス番号	用途
ワード	400839	過電圧ワーニング(上位)
	400841	不足電圧ワーニング(上位)
	400843	カレントオン時位置偏差過大ワーニング(上位)
	400897	電子ギア A(上位)
	400899	電子ギア B(上位)
	400901	モーター回転方向(上位)
	400903	ソフトウェアオーバートラベル(上位)
	400905	+ソフトウェアリミット(上位)
	400907	-ソフトウェアリミット(上位)
	400909	プリセット位置(上位)
	400911	ラウンド設定(上位)
	400913	ラウンド設定範囲(上位)
	400961	データ設定器速度表示(上位)
	400963	データ設定器表示(上位)
	400965	アブソリュートバックアップシステム(上位)
	401025+2n (n=0～63)	位置 No.0～63(上位)
	401153+2n (n=0～63)	運転速度 No.0～63(上位)
	401281+2n (n=0～63)	運転方式 No.0～63 (上位)
	401282+2n (n=0～63)	運転方式 No.0～63(下位)
	401409+2n (n=0～63)	運転機能 No.0～63(上位)
	401410+2n (n=0～63)	運転機能 No.0～63(下位)
	401537+2n (n=0～63)	加速 No.0～63(上位)
	401665+2n (n=0～63)	減速 No.0～63(上位)
	401793+2n (n=0～63)	押し当て電流 No.0～63(上位)
	401921+2n (n=0～63)	順送り位置決め No.0～63(上位)
	401922+2n (n=0～63)	順送り位置決め No.0～63(下位)
	402049+2n (n=0～63)	ドウェル時間 No.0～63(上位)
	404097+2n (n=0～5)	MS0 運転 No.選択(上位)～MS5 運転 No.選択(上位)
	404109	HOME-P 出力機能選択(上位)
	404129	フィルタ選択(上位)
	404131	速度差ゲイン 1(上位)
	404133	速度差ゲイン 2(上位)
	404135	制御モード(上位)
	404137	スムースドライブ(上位)
	404161	自動復帰動作(上位)
	404163	自動復帰運転速度(上位)
	404165	自動復帰加減速(上位)
	404167	自動復帰起動速度(上位)
	404169	JOG 運転(上位)
	404225	カレントオフ時位置偏差過大アラーム(上位)
	404353+2n (n=0～7)	IN0 入力機能選択(上位)～IN7 入力機能選択(上位)
	404354+2n (n=0～7)	IN0 入力機能選択(下位)～IN7 入力機能選択(下位)
	404385+2n (n=0～7)	IN0 入力接点設定(上位)～IN7 入力接点設定(上位)
	404417+2n (n=0～5)	OUT0 出力機能選択(上位)～OUT5 出力機能選択(上位)
	404418+2n (n=0～5)	OUT0 出力機能選択(下位)～OUT5 出力機能選択(下位)
	404449+2n (n=0～15)	NET-IN0 入力機能選択(上位)～NET-IN15 入力機能選択(上位)
	404450+2n (n=0～15)	NET-IN0 入力機能選択(下位)～NET-IN15 入力機能選択(下位)
	404481+2n (n=0～15)	NET-OUT0 出力機能選択(上位)～NET-OUT15 出力機能選択(上位)
	404482+2n (n=0～15)	NET-OUT0 出力機能選択(下位)～NET-OUT15 出力機能選択(下位)
	404609	通信タイムアウト(上位)
	404611	通信異常アラーム(下位)

5.4.2 GOT の内部デバイス

タイプ	デバイス番号	用途
ビット	GB40	スクリプトトリガ
	GB61000	状態フラグ
	GB61001	号機選択状態フラグ
	GB61002	電源タイプ状態フラグ
	GB61010	電源タイプ判別フラグ
	GB62000	表示運転データ上スクロール
	GB62001	表示運転データ下スクロール
	GB62002	運転データ入力の読込みトリガ
	GB62003	運転データ入力の書込みトリガ
	GB62004	I/O 機能選択の書込みトリガ
	GB62005	OUT/NET-OUT 区別ビット
	GB62006	オーバーラップウィンドウ表示中フラグ
	GB62007	画面遷移禁止インターロック用フラグ
	GB62008	マルチ運転中インターロック用フラグ
	GB62009	マルチ起動中・終了中フラグ
	GB62010	開始・終了前確認トリガ
	GB62011	アラーム発生状態確認フラグ
	GB62012	運転中確認フラグ
	GB62013	運転準備開始トリガ
	GB62014	運転準備中フラグ
	GB62015	タッチパネル運転中フラグ
	GB62016	運転終了開始トリガ
	GB62017	運転終了フラグ
	GB62018	I/O テストのフラグ
	GB62019	位置確定トリガ
	GB62020+10n (n=0～30)	開始・終了前確認トリガ(1 軸目～31 軸目)
	GB62021+10n (n=0～30)	アラーム発生状態確認フラグ(1 軸目～31 軸目)
	GB62022+10n (n=0～30)	運転中確認フラグ(1 軸目～31 軸目)
	GB62023+10n (n=0～30)	運転準備開始トリガ(1 軸目～31 軸目)
	GB62024+10n (n=0～30)	運転準備中フラグ(1 軸目～31 軸目)
	GB62025+10n (n=0～30)	タッチパネル運転中フラグ(1 軸目～31 軸目)
	GB62026+10n (n=0～30)	運転終了開始トリガ(1 軸目～31 軸目)
	GB62027+10n (n=0～30)	運転終了フラグ(1 軸目～31 軸目)
	GD60031.b0	アラームリセット
	GD61020.b0～b5	運転データ入力画面の運転番号の No.表示
	GD61200+100n.b0 (n=0～30)	運転番号の No.表示(1 軸目～31 軸目)
	GD61200+100n.b1 (n=0～30)	運転番号の No.表示(1 軸目～31 軸目)
	GD61200+100n.b2 (n=0～30)	運転番号の No.表示(1 軸目～31 軸目)
	GD61200+100n.b3 (n=0～30)	運転番号の No.表示(1 軸目～31 軸目)
	GD61200+100n.b4 (n=0～30)	運転番号の No.表示(1 軸目～31 軸目)
	GD61200+100n.b5 (n=0～30)	運転番号の No.表示(1 軸目～31 軸目)
	GS281.b0～GS282.b15	異常局情報(CH1)
	GS512.b0	時刻変更信号
	GS531.b0～GS532.b15	モニタ局切断(CH1)
ワード	GD60000	ベース画面切り換え
	GD60001	オーバーラップウィンドウ 1 画面切り換え
	GD60004	オーバーラップウィンドウ 2 画面切り換え
	GD60021	言語切り換え
	GD60022	システム言語切り換え
	GD60031、GD60041	システム情報
	GD60042	現在カーソル表示ユーザ ID

タイプ	デバイス番号	用途
ワード	GD60080～GD60082	ドキュメント表示、ページ番号、前頁スイッチ、次頁スイッチ
	GD61000	号機番号入力
	GD61001～GD61008	運転データの No.の番号表示(0 行目～7 行目)
	GD61010	上下スクロールカウント値
	GD61011	Modbus アドレスのオフセット値
	GD61012	運転データ No.のタッチ位置情報
	GD61015	ティーチング時の指令位置、最小移動量表示の Modbus アドレスオフセット値
	GD61020	運転データ入力画面の運転番号の No.表示
	GD61021	運転データ入力画面の運転方式
	GD61022	運転データ入力画面の位置
	GD61024	運転データ入力画面の運転速度
	GD61026	運転データ入力画面の運転機能
	GD61027	運転データ入力画面の押し当て電流
	GD61029	運転データ入力画面のドウェル時間
	GD61031	運転データ入力画面の順送り位置決め
	GD61032	運転データ入力画面の加速
	GD61034	運転データ入力画面の減速
	GD61040	IN 入力の信号番号
	GD61041	OUT 出力の信号番号
	GD61042	NET-IN 入力の信号番号
	GD61043	NET-OUT 出力の信号番号
	GD61044	IN 入力機能選択の一時保存デバイス
	GD61046	OUT/NET-OUT 出力機能選択の一時保存デバイス
	GD61048	NET-IN 入力機能選択の一時保存デバイス
	GD61050～GD61080	NET-IN0～NET-IN15 のバックアップデバイス
	GD61082～GD61096	IN0～IN7 のバックアップデバイス
	GD61098～GD61108	OUT0～OUT5 のバックアップデバイス
	GD61110	モニタの指令位置のバックアップデバイス
	GD61112	プリセット位置のバックアップデバイス
	GD61114	JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
	GD61116	Configuration 実行待ち時間用タイマー
	GD61199	局番バックアップ
	GD61200+100n(n=0～30)	運転番号の No.表示(1 軸目～31 軸目)
	GD61202+100n(n=0～30)	NET-IN0 のバックアップデバイス(1 軸目～31 軸目)
	GD61204+100n(n=0～30)	NET-IN1 のバックアップデバイス(1 軸目～31 軸目)
	GD61206+100n(n=0～30)	NET-IN2 のバックアップデバイス(1 軸目～31 軸目)
	GD61208+100n(n=0～30)	NET-IN3 のバックアップデバイス(1 軸目～31 軸目)
	GD61210+100n(n=0～30)	NET-IN4 のバックアップデバイス(1 軸目～31 軸目)
	GD61212+100n(n=0～30)	NET-IN5 のバックアップデバイス(1 軸目～31 軸目)
	GD61214+100n(n=0～30)	NET-IN6 のバックアップデバイス(1 軸目～31 軸目)
	GD61216+100n(n=0～30)	NET-IN7 のバックアップデバイス(1 軸目～31 軸目)
	GD61218+100n(n=0～30)	NET-IN8 のバックアップデバイス(1 軸目～31 軸目)
	GD61220+100n(n=0～30)	NET-IN9 のバックアップデバイス(1 軸目～31 軸目)
	GD61222+100n(n=0～30)	NET-IN10 のバックアップデバイス(1 軸目～31 軸目)
	GD61224+100n(n=0～30)	NET-IN11 のバックアップデバイス(1 軸目～31 軸目)
	GD61226+100n(n=0～30)	NET-IN12 のバックアップデバイス(1 軸目～31 軸目)
	GD61228+100n(n=0～30)	NET-IN13 のバックアップデバイス(1 軸目～31 軸目)
	GD61230+100n(n=0～30)	NET-IN14 のバックアップデバイス(1 軸目～31 軸目)
	GD61232+100n(n=0～30)	NET-IN15 のバックアップデバイス(1 軸目～31 軸目)
	GD61234+100n(n=0～30)	IN0 のバックアップデバイス(1 軸目～31 軸目)
	GD61236+100n(n=0～30)	IN1 のバックアップデバイス(1 軸目～31 軸目)
	GD61238+100n(n=0～30)	IN2 のバックアップデバイス(1 軸目～31 軸目)

タイプ	デバイス番号	用途
ワード	GD61240+100n(n=0～30)	IN3 のバックアップデバイス(1 軸目～31 軸目)
	GD61242+100n(n=0～30)	IN4 のバックアップデバイス(1 軸目～31 軸目)
	GD61244+100n(n=0～30)	IN5 のバックアップデバイス(1 軸目～31 軸目)
	GD61246+100n(n=0～30)	IN6 のバックアップデバイス(1 軸目～31 軸目)
	GD61248+100n(n=0～30)	IN7 のバックアップデバイス(1 軸目～31 軸目)
	GD61250+100n(n=0～30)	モニタの指令位置のバックアップデバイス(1 軸目～31 軸目)
	GD61252+100n(n=0～30)	プリセット位置のバックアップデバイス(1 軸目～31 軸目)
	GD61254+100n(n=0～30)	JOG 移動量のバックアップデバイス(1 軸目～31 軸目)
	GD61256+100n(n=0～30)	Configuration 実行待ち時間用タイマー(1 軸目～31 軸目)
	GD64990～GD64995	時計のデジスイッチ
	GD65000	号機番号
	GD65001	電源タイプ指定
	GD65100～GD65102	外部制御デバイス(レシピ)
	GD65103～GD65105	外部通知デバイス(レシピ)
	GS7	一秒バイナリカウンタ
	GS513～GS516	変更時刻
	GS531～GS532	モニタ局切断(CH1)
	GS650～GS652	現在時刻
	TMP950～TMP996	スクリプト演算用

5.5 コメント一覧

コメントグループ No.	コメント No.	使用箇所
497	No.1～240	B-31015、B-31018、B-31019
498	No.1～60	B-31012、B-31013、B-31016、B-31017、B-31020、W-32004、W-32005
499	No.1～90	B-31012、B-31013、B-31016、B-31017、B-31020、W-32006
500	No.1、2	B-30002～B-30006、B-30500～B-31030
	No.3	B-30002、B-30003
	No.4	B-30002、B-30004
	No.5	B-30002、B-30005
	No.6	B-30002、B-30006
	No.7	B-30002
	No.8	B-30001
	No.9～13	B-30003、B-31004～B-31008
	No.14～16、18	B-30004、B-31010～B-31014
	No.17	B-30004
	No.19～20	B-30005、B-30500～B-30502、B-31015～B-31019
	No.21～23	B-30005
	No.24～26	B-30006、B-31020～B-31029
	No.27	B-30003～B-30006、B-31002、B-31010～B-31014、B-31030
	No.28	B-30003～B-30006、B-31002～B-31008、B-31030
	No.29～31	B-30003～B-30006、B-31002、B-31030
	No.32～33	B-31004～B-31008
	No.34	B-31010～B-31014
	No.35、38～40	B-30500～B-30502、B-31015～B-31019
	No.36	B-30003～B-30005、B-30500～B-31019、B-31030
	No.37	B-30005、B-30500～B-30502
	No.50～51	B-30001～B-30006、B-31002～B-31021、B-31030
	No.100～128	B-31002
	No.150～185	B-31004
	No.200～218	B-31005
	No.250～269	B-31006
	No.300～316	B-31007
	No.350～362	B-31008

コメントグループ No.	コメント No.	使用箇所
500	No.400～413	B-31010
	No.450～457	B-31011
	No.500～524	B-31012
	No.550～582	B-31013
	No.600～602	B-31014
	No.650～660	B-31015
	No.700～702	B-31016
	No.750～752	B-31017
	No.800～806	B-31018
	No.850～856	B-31019
	No.900～905	B-31020
	No.950～966	B-31021
	No.1000～1014、1046	B-31022～B-31029
	No.1015～1018	B-31022
	No.1019～1022	B-31023
	No.1023～1026	B-31024
	No.1027～1030	B-31025
	No.1031～1034	B-31026
	No.1035～1038	B-31027
	No.1039～1042	B-31028
	No.1043～1045	B-31029
	No.1050～1064	B-31030
	No.1100～1101	W-30001
	No.1150	W-30002
	No.1200～1208	W-30003
	No.1250～1284	W-30004
	No.1300～1334	W-32001
	No.1350～1354	W-32002
	No.1400～1403	W-32003
	No.1450～1451	W-32004～W-32006
	No.1500～1503	W-32007
	No.1550～1552	W-32008

5.6 スクリプト一覧

項目	設定
プロジェクトスクリプト	有り
画面スクリプト	B-31002、B-31021～B-31029、W-30002、W-30004、W-32001、W-32004～W-32006
オブジェクトスクリプト	W-30003

5.6.1 プロジェクトスクリプト

スクリプトNo.	30001	スクリプト名	Script30001
コメント	初期設定		
データ形式	符号付きBIN16	トリガ種別	立上り GB40

```
[w:GD60080]=1;    //ベース画面30500のドキュメントページNo.を1に設定
[w:GD60081]=1;    //ベース画面30501のドキュメントページNo.を1に設定
[w:GD60082]=1;    //ベース画面30502のドキュメントページNo.を1に設定

[w:GD61000] = 1;    // スレーブ(ドライバ)号機番号の初期値(局番切り換えの番号)
```

スクリプトNo.	31001	スクリプト名	Script31001
コメント	運転データ画面のNoの初期値設定		
データ形式	符号付きBIN16	トリガ種別	立上り GB40

```
// 運転データ画面のNo.の初期値の設定

// プロジェクトスクリプト No.31001
// GB40(GOT常時ONビット)の立上りで、スクリプトを起動
// このスクリプトは、GOTの電源投入後、1回のみ実行する

[u16:GD61001] = 0;     // 0行目の初期値No.0
[u16:GD61002] = 1;     // 1行目の初期値No.1
[u16:GD61003] = 2;     // 2行目の初期値No.2
[u16:GD61004] = 3;     // 3行目の初期値No.3
[u16:GD61005] = 4;     // 4行目の初期値No.4
[u16:GD61006] = 5;     // 5行目の初期値No.5
[u16:GD61007] = 6;     // 6行目の初期値No.6
[u16:GD61008] = 7;     // 7行目の初期値No.7
```

スクリプトNo.	31002	スクリプト名	Script31002
コメント	接続機器の通信を全軸有効にする		
データ形式	符号付きBIN16	トリガ種別	周期 1(秒)

```
// 接続機器の全軸の通信接続を有効にするスクリプト
// マルチ運転は、局番切り換えが使えない(局番=ドライバの号機番号ごとに)為、
// 全軸の通信接続を切断する必要がある
// その為、マルチ運転画面からの復旧に対して、全軸の通信接続を有効(接続)する必要がある

// プロジェクトスクリプト No.31002
// 周期で、1秒サイクルのトリガで、スクリプト起動

// マルチ運転のベース画面以外は、全軸の通信接続を有効にする
if([w:GD60000] >= 30001 && [w:GD60000] <= 31021){
   [w:GS531] = 0;    //CH1の局番(ドライバの号機番号)0～15軸の通信を接続、0で接続(通信有効)、1で切
断
   [w:GS532] = 0;    // CH1の局番(ドライバの号機番号)16～31軸の通信を接続、0で接続(通信有効)、1で
切断
}

// SYSTEMのベース画面も、全軸の通信接続を有効にする
```

```
if([w:GD60000] == 31030){
   [w:GS531] = 0;
   [w:GS532] = 0;
}
```

スクリプト No.	31014	スクリプト名	Script31014
コメント	インターロック		
データ形式	符号付き BIN16	トリガ種別	常時

```
// 画面切り換えのインターロックをかけるスクリプト
// オーバーラップウィンドウが表示中や運転準備中、運転中は画面切り換えができないようインターロック
フラグを立てる

// プロジェクトスクリプト No.31014
// 常時 ON で、スクリプトを起動

// [b:GB62006] : オーバーラップウィンドウ表示中フラグ
// [b:GB62007] : 画面切り換えインターロック
// [b:GB62010] : 開始_終了前確認フラグ
// [b:GB62013] : 運転準備中フラグ
// [b:GB62015] : タッチパネル運転中フラグ
// [b:GB62016] : 運転終了開始トリガ

if([w:GD60004] != 0){ // オーバーラップウィンドウが表示中の場合
   [b:GB62006] = 1; // オーバーラップウィンドウ表示中フラグを ON
} else {             // オーバーラップウィンドウが表示されていない場合
   [b:GB62006] = 0; // オーバーラップウィンドウ表示中フラグのクリア
}

// オーバーラップウィンドウ表示中、開始_終了前確認、運転準備中、タッチパネル運転中、運転終了中の
// いずれかが発生した場合は、画面切り換えをできないように、インターロックフラグを立てる
[b:GB62007] = [b:GB62006] | [b:GB62010] | [b:GB62013] | [b:GB62015] | [b:GB62016];
```

5.6.2 画面スクリプト

ベース画面 31002

スクリプト No.	31003	スクリプト名	Script31003
コメント	上方向の表示		
データ形式	符号付き BIN16	トリガ種別	立上り GB62000

```
// 運転データの表示を上下にスクロールする
// 上下スクロールカウンタの上方向カウント数を求める

// 運転データ画面スクリプト スクリプト No.31003
// GB62000 の立上りで、スクリプトを起動

[w:GD61010] = [w:GD61010] - 1;           // 上下スクロールカウント値
[w:GD61011] = [w:GD61011] - 16;          // Modbus アドレスのオフセット値

if ([w:GD61010] < 0) { [w:GD61010] = 0; }    // No.0 を表示中に、上方向ボタンをタッチしても変わらな
いようにロックする
if ([w:GD61011] < 0) { [w:GD61011] = 0; }    // Modbus オフセットも、同様にロックする

[b:GB62000] = OFF;
```

スクリプト No.	31004	スクリプト名	Script31004
コメント	下方向の表示		
データ形式	符号付き BIN16	トリガ種別	立上り GB62001

```
// 運転データの表示を上下にスクロールする
// 上下スクロールカウンタの下方向カウント数を求める
```

```
// 運転データ画面スクリプト スクリプトNo.31004
// GB62001の立上りで、スクリプトを起動

[w:GD61010] = [w:GD61010] + 1;            // 上下スクロールカウント値
[w:GD61011] = [w:GD61011] + 16;           // Modbusアドレスのオフセット値

if ([w:GD61010] >= 8) { [w:GD61010] = 7; }   // 下方向ボタンを 8 回以上押しても、No.63 以降を表示させない
if ([w:GD61011] >= 128) { [w:GD61011] = 112; }    // 同上(16×7=112 の Modbus オフセット、No.56 が基点になる)

[b:GB62001] = OFF;
```

スクリプトNo.	31013	スクリプト名	Script31013
コメント	位置確定		
データ形式	符号付きBIN16	トリガ種別	立上り GB62019

```
// ティーチングの位置確定
// 運転データ画面のティーチングで位置確定を行うスクリプト

// 位置確定スクリプト No.31013
// 位置確定ボタン(GB62019:ビットモーメンタリ)の立上りで、スクリプトを起動

// [u16:GD61015] : ティーチング時の指令位置の表示のModbusオフセット値


// 選択した運転データNo.の位置に、モニタの指令位置を書き込む
// もし、モニタの指令位置が、設定範囲を超えていた場合は、Max or Minで固定する
if ([s32:400199] <= -8388608 || [s32:400199] >= 8388607) {
  if ([s32:400199] <= -8388608) {
    [s32:401025[u16:GD61015]] = -8388608;   // Minでクリップする
  }
  if ([s32:400199] >= 8388607) {
    [s32:401025[u16:GD61015]] = 8388607;      // Maxでクリップする
  }
} else {
    [s32:401025[u16:GD61015]] = [s32:400199];    // 設定範囲内の場合
}

// 選択した運転データNo.の運転方式をアブソリュート(ABS)に設定する
[s32:401281[u16:GD61015]] = 1;
```

スクリプトNo.	31005	スクリプト名	Script31005
コメント	運転データ入力_読込み		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62002

```
// 運転データ入力のオーバーラップウィンドウに、値を読み出して表示する

// 運転データ画面スクリプト スクリプト No.31005
// 運転データ画面のGB62002のON中で、スクリプトを起動

// [b:GB62002] : 本スクリプトの起動トリガ、ビットセットで起動
// [w:GD61010] : 上下スクロールカウント値
// [w:GD61011] : Modbusアドレスのオフセット値
// [w:GD61012] : 運転データNo.のタッチ位置情報(何行目)
// [u16:GD61020] : 運転番号のNo.の値の表示

[u16:GD61020] = [w:GD61012] + ([w:GD61010] * 8);       // 運転番号のNo.の値を算出

// 何行目でタッチされたかで、データNo.何番かの運転データ値を振り分けて、ウィンドウに表示する
```

```
switch ([w:GD61012])
{
  case 0 : [u16:GD61021]= [u16:401282[w:GD61011]];    // 0 行目の運転方式
        [s32:GD61022] = [s32:401025[w:GD61011]];    // 0 行目の位置
        [u32:GD61024] = [u32:401153[w:GD61011]];    // 0 行目の運転速度
        [u16:GD61026] = [u16:401410[w:GD61011]];    // 0 行目の運転機能
        [u32:GD61027] = [u32:401793[w:GD61011]];    // 0 行目の押し当て電流
        [u32:GD61029] = [u32:402049[w:GD61011]];    // 0 行目のドウェル時間
        [u16:GD61031] = [u16:401922[w:GD61011]];    // 0 行目の順送り位置決め
        [u32:GD61032] = [u32:401537[w:GD61011]];    // 0 行目の加速
        [u32:GD61034] = [u32:401665[w:GD61011]];    // 0 行目の減速
        break;

  case 1 : [u16:GD61021]= [u16:401284[w:GD61011]];    // 1 行目の運転方式
        [s32:GD61022] = [s32:401027[w:GD61011]];    // 1 行目の位置
        [u32:GD61024] = [u32:401155[w:GD61011]];    // 1 行目の運転速度
        [u16:GD61026] = [u16:401412[w:GD61011]];    // 1 行目の運転機能
        [u32:GD61027] = [u32:401795[w:GD61011]];    // 1 行目の押し当て電流
        [u32:GD61029] = [u32:402051[w:GD61011]];    // 1 行目のドウェル時間
        [u16:GD61031] = [u16:401924[w:GD61011]];    // 1 行目の順送り位置決め
        [u32:GD61032] = [u32:401539[w:GD61011]];    // 1 行目の加速
        [u32:GD61034] = [u32:401667[w:GD61011]];    // 1 行目の減速
        break;

  case 2 : [u16:GD61021]= [u16:401286[w:GD61011]];    // 2 行目の運転方式
        [s32:GD61022] = [s32:401029[w:GD61011]];    // 2 行目の位置
        [u32:GD61024] = [u32:401157[w:GD61011]];    // 2 行目の運転速度
        [u16:GD61026] = [u16:401414[w:GD61011]];    // 2 行目の運転機能
        [u32:GD61027] = [u32:401797[w:GD61011]];    // 2 行目の押し当て電流
        [u32:GD61029] = [u32:402053[w:GD61011]];    // 2 行目のドウェル時間
        [u16:GD61031] = [u16:401926[w:GD61011]];    // 2 行目の順送り位置決め
        [u32:GD61032] = [u32:401541[w:GD61011]];    // 2 行目の加速
        [u32:GD61034] = [u32:401669[w:GD61011]];    // 2 行目の減速
        break;

  case 3 : [u16:GD61021]= [u16:401288[w:GD61011]];    // 3 行目の運転方式
        [s32:GD61022] = [s32:401031[w:GD61011]];    // 3 行目の位置
        [u32:GD61024] = [u32:401159[w:GD61011]];    // 3 行目の運転速度
        [u16:GD61026] = [u16:401416[w:GD61011]];    // 3 行目の運転機能
        [u32:GD61027] = [u32:401799[w:GD61011]];    // 3 行目の押し当て電流
        [u32:GD61029] = [u32:402055[w:GD61011]];    // 3 行目のドウェル時間
        [u16:GD61031] = [u16:401928[w:GD61011]];    // 3 行目の順送り位置決め
        [u32:GD61032] = [u32:401543[w:GD61011]];    // 3 行目の加速
        [u32:GD61034] = [u32:401671[w:GD61011]];    // 3 行目の減速
        break;

  case 4 : [u16:GD61021]= [u16:401290[w:GD61011]];    // 4 行目の運転方式
        [s32:GD61022] = [s32:401033[w:GD61011]];    // 4 行目の位置
        [u32:GD61024] = [u32:401161[w:GD61011]];    // 4 行目の運転速度
        [u16:GD61026] = [u16:401418[w:GD61011]];    // 4 行目の運転機能
        [u32:GD61027] = [u32:401801[w:GD61011]];    // 4 行目の押し当て電流
        [u32:GD61029] = [u32:402057[w:GD61011]];    // 4 行目のドウェル時間
        [u16:GD61031] = [u16:401930[w:GD61011]];    // 4 行目の順送り位置決め
        [u32:GD61032] = [u32:401545[w:GD61011]];    // 4 行目の加速
        [u32:GD61034] = [u32:401673[w:GD61011]];    // 4 行目の減速
        break;
```

```
  case 5 : [u16:GD61021]= [u16:401292[w:GD61011]];    // 5 行目の運転方式
        [s32:GD61022] = [s32:401035[w:GD61011]];    // 5 行目の位置
        [u32:GD61024] = [u32:401163[w:GD61011]];    // 5 行目の運転速度
        [u16:GD61026] = [u16:401420[w:GD61011]];    // 5 行目の運転機能
        [u32:GD61027] = [u32:401803[w:GD61011]];    // 5 行目の押し当て電流
        [u32:GD61029] = [u32:402059[w:GD61011]];    // 5 行目のドウェル時間
        [u16:GD61031] = [u16:401932[w:GD61011]];    // 5 行目の順送り位置決め
        [u32:GD61032] = [u32:401547[w:GD61011]];    // 5 行目の加速
        [u32:GD61034] = [u32:401675[w:GD61011]];    // 5 行目の減速
        break;

  case 6 : [u16:GD61021]= [u16:401294[w:GD61011]];    // 6 行目の運転方式
        [s32:GD61022] = [s32:401037[w:GD61011]];    // 6 行目の位置
        [u32:GD61024] = [u32:401165[w:GD61011]];    // 6 行目の運転速度
        [u16:GD61026] = [u16:401422[w:GD61011]];    // 6 行目の運転機能
        [u32:GD61027] = [u32:401805[w:GD61011]];    // 6 行目の押し当て電流
        [u32:GD61029] = [u32:402061[w:GD61011]];    // 6 行目のドウェル時間
        [u16:GD61031] = [u16:401934[w:GD61011]];    // 6 行目の順送り位置決め
        [u32:GD61032] = [u32:401549[w:GD61011]];    // 6 行目の加速
        [u32:GD61034] = [u32:401677[w:GD61011]];    // 6 行目の減速
        break;

  case 7 : [u16:GD61021]= [u16:401296[w:GD61011]];    // 7 行目の運転方式
        [s32:GD61022] = [s32:401039[w:GD61011]];    // 7 行目の位置
        [u32:GD61024] = [u32:401167[w:GD61011]];    // 7 行目の運転速度
        [u16:GD61026] = [u16:401424[w:GD61011]];    // 7 行目の運転機能
        [u32:GD61027] = [u32:401807[w:GD61011]];    // 7 行目の押し当て電流
        [u32:GD61029] = [u32:402063[w:GD61011]];    // 7 行目のドウェル時間
        [u16:GD61031] = [u16:401936[w:GD61011]];    // 7 行目の順送り位置決め
        [u32:GD61032] = [u32:401551[w:GD61011]];    // 7 行目の加速
        [u32:GD61034] = [u32:401679[w:GD61011]];    // 7 行目の減速
        break;
}

[b:GB62002] = 0;    // このスクリプトの起動トリガを終了する
```

ベース画面 31002、31021

スクリプト No.	31008	スクリプト名	Script31008
コメント	テスト運転時指令位置表示用		
データ形式	符号付き BIN16	トリガ種別	常時

```
// 運転データのティーチングと TEST の運転テスト時の指令位置を表示する

// 運転データ画面、TEST の運転画面 No.31008
// 常時 ON で、スクリプトを起動

// [u16:GD61020] : 運転番号の No.の値の表示

[u16:GD61015] = [u16:GD61020] * 2; // ティーチング時の指令位置の表示の Modbus オフセット値
```

スクリプト No.	31011	スクリプト名	Script31011
コメント	運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62015

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データ No.を M0～M5(割付直した NET-IN)に反映させる
// 反映した M0～M5 は、ドライバ入力指令(400126)にて、ON する

// 運転中スクリプト No.31011
// 運転準備スクリプト(No.31010)の GB62015 の ON 中で、スクリプトを起動

// [b:GB62015] : タッチパネル運転中フラグ(このスクリプトの起動トリガ)
//               タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット
// [u16:400196]  : モニタの現在の選択データ No.の Modbus アドレス
// [u16:GD61020] : 運転データ No.のデバイス


// タッチパネル運転中の場合
if([b:GB62015] == 1){

   // 選択した運転データ No.をドライバ入力指令の M0～M5 に反映させる
   // タッチパネルの処理負荷を軽減させる為に、
   // モニタの現在の選択データ No.(400196)と選択した運転データ No.を
   // 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

   if([u16:400196] != [u16:GD61020]){
      [b:400126.b0] = [b:GD61020.b0];    // M0
      [b:400126.b1] = [b:GD61020.b1];    // M1
      [b:400126.b2] = [b:GD61020.b2];    // M2
      [b:400126.b3] = [b:GD61020.b3];    // M3
      [b:400126.b4] = [b:GD61020.b4];    // M4
      [b:400126.b5] = [b:GD61020.b5];    // M5
   }
}
```

ベース画面 31002、31020、31021

スクリプト No.	31009	スクリプト名	Script31009
コメント	開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62010

```
// 運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 運転データ画面、TEST の I/O テスト、運転(単独の運転)の画面スクリプト

// 開始_終了前確認スクリプト No.31009
// それぞれのベース画面の開始ボタンの ON 中周期、2 秒で、スクリプトを起動
```

```
// [b:GB62010] : 本スクリプトの起動トリガ、ビットセットで起動
// [b:GB62015] : タッチパネル運転中フラグ

// アラームチェック
if([s32:400129] == 0){      // アラームが発生していない場合
   [b:GB62011] = 1;       // アラーム発生状態確認フラグ(1:アラーム発生なし)
   [w:GD60004] = 0;       // アラーム発生中確認オーバーラップウィンドウの消去
} else {                  // アラームが発生している場合
   [b:GB62011] = 0;       // アラーム発生状態確認フラグ(0:アラーム発生中)
   [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
}

// 運転中チェック
if([s32:400201] == 0){      // 運転していない場合
   [b:GB62012] = 1;       // 運転中確認フラグ(1:運転していない)
} else {                  // 運転中の場合
   [b:GB62012] = 0;       // 運転中確認フラグ(0:運転中)
   [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
}

// 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
if([b:GB62011] == 1 && [b:GB62012] == 1){
   if([b:GB62015] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62010] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62013] = 1;     // 運転準備開始トリガ
      [b:GB62011] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62012] = 0;     // 運転中確認フラグのクリア
   } else {               // タッチパネル運転中なら、運転終了準備をする
      [b:GB62010] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62016] = 1;     // 運転終了開始トリガ
      [b:GB62011] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62012] = 0;     // 運転中確認フラグのクリア
   }
} else {
      [b:GB62010] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62011] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62012] = 0;     // 運転中確認フラグのクリア
}
```

スクリプト No.	31010	スクリプト名	Script31010
コメント	運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62013

```
// タッチパネルから運転できるように準備するスクリプト
// NET-IN などの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 運転データ画面、TEST の I/O テスト、運転(単独の運転)の画面スクリプト

// 運転準備スクリプト No.31010
// 開始_終了前確認スクリプト(No.31009)の GB62013 の ON 中で、スクリプトを起動

// [b:GB62013] : 運転準備開始トリガ(このスクリプトの起動トリガ)
// [b:GB62014] : 運転準備中フラグ
// [b:GB62015] : タッチパネル運転中フラグ
// [b:GB62018] : I/O テストのフラグ
// [u32:GD61050] : NET-IN0～15 のバックアップデバイス
// [u32:GD61082] : IN0～7 のバックアップデバイス
// [u32:GD61098] : OUT0～5 のバックアップデバイス
```

```
// [s32:GD61110]  :  モニタの指令位置のバックアップデバイス
// [s32:GD61112]  :  プリセット位置のバックアップデバイス
// [u32:GD61114]  :  JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD61116]  :  Configuration 実行待ち時間用タイマー


// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62013] == 1 && [b:GB62014] == 0){

   [b:GB62014] = 1;               // 運転準備中フラグ(1:運転準備中)
   bmov([u32:404449],[u32:GD61050],16); // NET-IN0～15 のバックアップ
   bmov([u32:404353],[u32:GD61082],8);  // IN0～7 のバックアップ
   bmov([u32:404417],[u32:GD61098],6);  // OUT0～5 のバックアップ
   [s32:GD61112] = [s32:400909];      // プリセット位置のバックアップ
   [u32:GD61114] = [u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

   // NET-IN の入力値をクリアする
   [w:400126] = 0;

   // タッチパネルから運転する為に信号の割当を変更する
   [u32:404461] = 4;// NET-IN6  START
   [u32:404463] = 3;// NET-IN7  HOME
   [u32:404465] = 18;  // NET-IN8  STOP
   [u32:404467] = 9;// NET-IN9  MS1
   [u32:404469] = 10; // NET-IN10 MS2
   [u32:404471] = 5;// NET-IN11 SSTART
   [u32:404473] = 6;// NET-IN12 +JOG
   [u32:404475] = 7;// NET-IN13 -JOG
   [u32:404477] = 1;// NET-IN14 FWD
   [u32:404479] = 2;// NET-IN15 RVS

   if([b:GB62018] == 0){ // 運転データ画面、TEST の運転画面の場合
      [u32:404449] = 48;  // NET-IN0  M0
      [u32:404451] = 49;  // NET-IN1  M1
      [u32:404453] = 50;  // NET-IN2  M2
      [u32:404455] = 51;  // NET-IN3  M3
      [u32:404457] = 52;  // NET-IN4  M4
      [u32:404459] = 53;  // NET-IN5  M5

      [u32:404353] = 32;  // IN0 R0
      [u32:404355] = 33;  // IN1 R1
      [u32:404357] = 34;  // IN2 R2
      [u32:404359] = 35;  // IN3 R3
      [u32:404361] = 36;  // IN4 R4
      [u32:404363] = 37;  // IN5 R5
      [u32:404365] = 18;  // IN6 STOP
      [u32:404367] = 39;  // IN7 R7

   } else {          // TEST の I/O テストの場合
      [u32:404449] = 40;  // NET-IN0  R8
      [u32:404451] = 41;  // NET-IN1  R9
      [u32:404453] = 42;  // NET-IN2  R10
      [u32:404455] = 43;  // NET-IN3  R11
      [u32:404457] = 44;  // NET-IN4  R12
      [u32:404459] = 45;  // NET-IN5  R13
```

```
      // IN/OUTについては、シミュレーションで信号名を表示させる為に、u16にする
      // シミュレーション上では、32ビット長の上位下位の反転ができないため。
      [u16:404354] = 32;  // IN0 R0
      [u16:404356] = 33;  // IN1 R1
      [u16:404358] = 34;  // IN2 R2
      [u16:404360] = 35;  // IN3 R3
      [u16:404362] = 36;  // IN4 R4
      [u16:404364] = 37;  // IN5 R5
      [u16:404366] = 38;  // IN6 R6
      [u16:404368] = 39;  // IN7 R7

      [u16:404418] = 40;  // OUT0 R8
      [u16:404420] = 41;  // OUT1 R9
      [u16:404422] = 42;  // OUT2 R10
      [u16:404424] = 43;  // OUT3 R11
      [u16:404426] = 44;  // OUT4 R12
      [u16:404428] = 45;  // OUT5 R13
   }

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD61110] = [s32:400199];
   [s32:400909] = [s32:GD61110];

   // Configurationの実行
   [s32:400397] = 1;

   // Configuration実行待ち時間のタイマー記録(単位は秒)
   [w:GD61116] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61116] > 1 && [b:GB62014] == 1){

   [s32:400395] = 1;        // P-PRESET(プリセット)実行する
   [s32:400909] = [s32:GD61112]; // バックアップしたプリセット位置パラメータを元に戻す
   [s32:400397] = 0;        // Configuration実行のゼロクリア
   [s32:400395] = 0;        // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62013] = 0;         // 運転準備開始トリガのクリア
                       // (このスクリプトの起動トリガを終了する)
   [b:GB62014] = 0;         // 運転準備中フラグのクリア
   [b:GB62015] = 1;         // タッチパネル運転中フラグ(1:運転中)
   [b:GB62018] = 0;         // I/Oテストのフラグのクリア
}
```

スクリプトNo.	31012	スクリプト名	Script31012
コメント	運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62016

```
// タッチパネルからの運転を終了するスクリプト
// バックアップしたNET-INなどの信号を元に戻したり、必要なパラメータなどを元に戻す
// 運転データ画面、TESTのI/Oテスト、運転(単独の運転)の画面スクリプト

// 運転終了スクリプト No.31012
// 開始_終了前確認スクリプト(No.31009)のGB62016の中で、スクリプトを起動

// [b:GB62015] : タッチパネル運転中フラグ
// [b:GB62016] : 運転終了開始トリガ(このスクリプトの起動トリガ)
```

```
// [b:GB62017] : 運転終了中フラグ
// [b:GB62018] : I/O テストのフラグ
// [u32:GD61050] : NET-IN0～15 のバックアップデバイス
// [u32:GD61082] : IN0～7 のバックアップデバイス
// [u32:GD61098] : OUT0～5 のバックアップデバイス
// [s32:GD61110] : モニタの指令位置のバックアップデバイス
// [s32:GD61112] : プリセット位置のバックアップデバイス
// [u32:GD61114] : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD61116] : Configuration 実行待ち時間用タイマー


// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62016] == 1 && [b:GB62017] == 0){

  [b:GB62015] = 0;              // タッチパネル運転中フラグのクリア
  [b:GB62017] = 1;              // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD61050],[u32:404449],16); // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD61082],[u32:404353],8);  // バックアップした IN0～7 を元に戻す
  bmov([u32:GD61098],[u32:404417],6);  // バックアップした OUT0～5 を元に戻す
  [u32:404169] = [u32:GD61114];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61110] = [s32:400199];
  [s32:400909] = [s32:GD61110];

  // TEST の I/O テスト画面で、出力テストをする際に、操作した NET-IN の入力値をクリアする
  [w:400126] = 0;

  // Configuration の実行
  [s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61116] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61116] > 1 && [b:GB62017] == 1){

  [s32:400395] = 1;        // P-PRESET(プリセット)実行する
  [s32:400909] = [s32:GD61112]; // プリセット位置パラメータを元に戻す
  [s32:400397] = 0;        // Configuration 実行のゼロクリア
  [s32:400395] = 0;        // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62016] = 0;         // 運転終了開始トリガのクリア
                        // (このスクリプトの起動トリガを終了する)
  [b:GB62017] = 0;         // 運転終了中フラグのクリア
  [b:GB62018] = 0;         // I/O テストのフラグのクリア
}
```

ベース画面 31022～31029

スクリプト No.	31015	スクリプト名	Script31015
コメント	マルチ運転インターロック		
データ形式	符号付き BIN16	トリガ種別	常時

```
// マルチ運転の準備中、終了中、オーバーラップウィンドウ表示中に他画面に移行しないようにインターロックをかける
// 1-31 軸のそれぞれのマルチ運転の画面スクリプト
```

```
// マルチ運転インターロックスクリプト No.31015
// 画面スクリプトの常時 ON で、スクリプトを起動

// [b:GB62020] ～ [b:GB62320] ： 1-31 軸のそれぞれのマルチ運転の起動トリガ、ビットセットで起動
// [b:GB62023] ～ [b:GB62323] ： 1-31 軸のそれぞれのマルチ運転準備開始トリガ
// [b:GB62026] ～ [b:GB62326] ： 1-31 軸のそれぞれのマルチ運転終了開始トリガ
// [b:GB62006]                ： オーバーラップウィンドウ表示中フラグ
// [b:GB62009]                ： マルチ運転準備・終了中インターロックフラグ

[b:GB62009] = [b:GB62006] | [b:GB62020] | [b:GB62023] | [b:GB62026] | [b:GB62030] | [b:GB62033] | [b:GB62036];
[b:GB62009] = [b:GB62009] | [b:GB62040] | [b:GB62043] | [b:GB62046] | [b:GB62050] | [b:GB62053] | [b:GB62056];
[b:GB62009] = [b:GB62009] | [b:GB62060] | [b:GB62063] | [b:GB62066] | [b:GB62070] | [b:GB62073] | [b:GB62076];
[b:GB62009] = [b:GB62009] | [b:GB62080] | [b:GB62083] | [b:GB62086] | [b:GB62090] | [b:GB62093] | [b:GB62096];
[b:GB62009] = [b:GB62009] | [b:GB62100] | [b:GB62103] | [b:GB62106] | [b:GB62110] | [b:GB62113] | [b:GB62116];
[b:GB62009] = [b:GB62009] | [b:GB62120] | [b:GB62123] | [b:GB62126] | [b:GB62130] | [b:GB62133] | [b:GB62136];
[b:GB62009] = [b:GB62009] | [b:GB62140] | [b:GB62143] | [b:GB62146] | [b:GB62150] | [b:GB62153] | [b:GB62156];
[b:GB62009] = [b:GB62009] | [b:GB62160] | [b:GB62163] | [b:GB62166] | [b:GB62170] | [b:GB62173] | [b:GB62176];
[b:GB62009] = [b:GB62009] | [b:GB62180] | [b:GB62183] | [b:GB62186] | [b:GB62190] | [b:GB62193] | [b:GB62196];
[b:GB62009] = [b:GB62009] | [b:GB62200] | [b:GB62203] | [b:GB62206] | [b:GB62210] | [b:GB62213] | [b:GB62216];
[b:GB62009] = [b:GB62009] | [b:GB62220] | [b:GB62223] | [b:GB62226] | [b:GB62230] | [b:GB62233] | [b:GB62236];
[b:GB62009] = [b:GB62009] | [b:GB62240] | [b:GB62243] | [b:GB62246] | [b:GB62250] | [b:GB62253] | [b:GB62256];
[b:GB62009] = [b:GB62009] | [b:GB62260] | [b:GB62263] | [b:GB62266] | [b:GB62270] | [b:GB62273] | [b:GB62276];
[b:GB62009] = [b:GB62009] | [b:GB62280] | [b:GB62283] | [b:GB62286] | [b:GB62290] | [b:GB62293] | [b:GB62296];
[b:GB62009] = [b:GB62009] | [b:GB62300] | [b:GB62303] | [b:GB62306] | [b:GB62310] | [b:GB62313] | [b:GB62316];
[b:GB62009] = [b:GB62009] | [b:GB62320] | [b:GB62323] | [b:GB62326];

// マルチ運転中にマルチ運転画面以外に移行しないようにインターロックをかける
// [b:GB62025] ～ [b:GB62325] ： 1-31 軸のそれぞれのマルチ運転中フラグ
// [b:GB62008]                ： マルチ運転中インターロックフラグ

[b:GB62008] = [b:GB62009] | [b:GB62025] | [b:GB62035] | [b:GB62045] | [b:GB62055] | [b:GB62065] | [b:GB62075];
[b:GB62008] = [b:GB62008] | [b:GB62085] | [b:GB62095] | [b:GB62105] | [b:GB62115] | [b:GB62125] | [b:GB62135];
[b:GB62008] = [b:GB62008] | [b:GB62145] | [b:GB62155] | [b:GB62165] | [b:GB62175] | [b:GB62185] | [b:GB62195];
[b:GB62008] = [b:GB62008] | [b:GB62205] | [b:GB62215] | [b:GB62225] | [b:GB62235] | [b:GB62245] | [b:GB62255];
[b:GB62008] = [b:GB62008] | [b:GB62265] | [b:GB62275] | [b:GB62285] | [b:GB62295] | [b:GB62305] | [b:GB62315];
```

```
[b:GB62008] = [b:GB62008] | [b:GB62325];

// 接続先異常の場合、マルチ運転・終了開始フラグをクリアする
// [w:GS281]、[w:GS282] ： 局番異常デバイス

if ([b:GS281.b1] == 1) { [b:GB62020] = 0; }
if ([b:GS281.b2] == 1) { [b:GB62030] = 0; }
if ([b:GS281.b3] == 1) { [b:GB62040] = 0; }
if ([b:GS281.b4] == 1) { [b:GB62050] = 0; }
if ([b:GS281.b5] == 1) { [b:GB62060] = 0; }
if ([b:GS281.b6] == 1) { [b:GB62070] = 0; }
if ([b:GS281.b7] == 1) { [b:GB62080] = 0; }
if ([b:GS281.b8] == 1) { [b:GB62090] = 0; }
if ([b:GS281.b9] == 1) { [b:GB62100] = 0; }
if([b:GS281.b10] == 1) { [b:GB62110] = 0; }
if ([b:GS281.b11] == 1)  { [b:GB62120] = 0; }
if ([b:GS281.b12] == 1)  { [b:GB62130] = 0; }
if ([b:GS281.b13] == 1)  { [b:GB62140] = 0; }
if ([b:GS281.b14] == 1)  { [b:GB62150] = 0; }
if ([b:GS281.b15] == 1)  { [b:GB62160] = 0; }

if ([b:GS282.b0] == 1) { [b:GB62170] = 0; }
if ([b:GS282.b1] == 1) { [b:GB62180] = 0; }
if ([b:GS282.b2] == 1) { [b:GB62190] = 0; }
if ([b:GS282.b3] == 1) { [b:GB62200] = 0; }
if ([b:GS282.b4] == 1) { [b:GB62210] = 0; }
if ([b:GS282.b5] == 1) { [b:GB62220] = 0; }
if ([b:GS282.b6] == 1) { [b:GB62230] = 0; }
if ([b:GS282.b7] == 1) { [b:GB62240] = 0; }
if ([b:GS282.b8] == 1) { [b:GB62250] = 0; }
if ([b:GS282.b9] == 1) { [b:GB62260] = 0; }
if ([b:GS282.b10] == 1)  { [b:GB62270] = 0; }
if ([b:GS282.b11] == 1)  { [b:GB62280] = 0; }
if ([b:GS282.b12] == 1)  { [b:GB62290] = 0; }
if ([b:GS282.b13] == 1)  { [b:GB62300] = 0; }
if ([b:GS282.b14] == 1)  { [b:GB62310] = 0; }
if ([b:GS282.b15] == 1)  { [b:GB62320] = 0; }
```

ベース画面 31022

スクリプト No.	31016	スクリプト名	Script31016
コメント	1-4_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62020

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 1-4 軸のマルチ運転の画面スクリプト

// 1-4_マルチ運転開始_終了前確認スクリプト No.31016
// 1-4 軸のマルチ運転画面のそれぞれの開始ボタンの ON 中周期、2 秒で、スクリプトを起動

// [b:GB62020] ： 1 軸目の起動トリガ、ビットセットで起動
// [b:GB62030] ： 2 軸目の起動トリガ、ビットセットで起動
// [b:GB62040] ： 3 軸目の起動トリガ、ビットセットで起動
// [b:GB62050] ： 4 軸目の起動トリガ、ビットセットで起動
// [b:GB62025] ： 1 軸目のタッチパネル運転中フラグ
// [b:GB62035] ： 2 軸目のタッチパネル運転中フラグ
// [b:GB62045] ： 3 軸目のタッチパネル運転中フラグ
// [b:GB62055] ： 4 軸目のタッチパネル運転中フラグ
```

```
// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 1 軸目
if([b:GB62020] == 1){
  [w:GD61000] = 1;

  // アラームチェック
  if([1-1:s32:400129] == 0){    // アラームが発生していない場合
    [b:GB62021] = 1;         // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;         // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                   // アラームが発生している場合
    [b:GB62021] = 0;         // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-1:s32:400201] == 0){    // 運転していない場合
    [b:GB62022] = 1;         // 運転中確認フラグ(1:運転していない)
  } else {                   // 運転中の場合
    [b:GB62022] = 0;         // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62021] == 1 && [b:GB62022] == 1){
    if([b:GB62025] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62020] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62023] = 1;     // 運転準備開始トリガ
      [b:GB62021] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62022] = 0;     // 運転中確認フラグのクリア
    } else {                 // タッチパネル運転中なら、運転終了準備をする
      [b:GB62020] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62026] = 1;     // 運転終了開始トリガ
      [b:GB62021] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62022] = 0;     // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62020] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62021] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62022] = 0;     // 運転中確認フラグのクリア
  }
}

// 2 軸目
if([b:GB62030] == 1){
  [w:GD61000] = 2;

  // アラームチェック
  if([1-2:s32:400129] == 0){    // アラームが発生していない場合
    [b:GB62031] = 1;         // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;         // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                   // アラームが発生している場合
```

```
    [b:GB62031] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-2:s32:400201] == 0){     // 運転していない場合
    [b:GB62032] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62032] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62031] == 1 && [b:GB62032] == 1){
    if([b:GB62035] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62030] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62033] = 1;     // 運転準備開始トリガ
      [b:GB62031] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62032] = 0;     // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
      [b:GB62030] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62036] = 1;     // 運転終了開始トリガ
      [b:GB62031] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62032] = 0;     // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62030] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62031] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62032] = 0;     // 運転中確認フラグのクリア
  }
}

// 3 軸目
if([b:GB62040] == 1){
  [w:GD61000] = 3;

  // アラームチェック
  if([1-3:s32:400129] == 0){     // アラームが発生していない場合
    [b:GB62041] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62041] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-3:s32:400201] == 0){     // 運転していない場合
    [b:GB62042] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62042] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62041] == 1 && [b:GB62042] == 1){
    if([b:GB62045] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
```

```
        [b:GB62040] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62043] = 1;    // 運転準備開始トリガ
        [b:GB62041] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62042] = 0;    // 運転中確認フラグのクリア
      } else {              // タッチパネル運転中なら、運転終了準備をする
        [b:GB62040] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62046] = 1;    // 運転終了開始トリガ
        [b:GB62041] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62042] = 0;    // 運転中確認フラグのクリア
      }
  } else {
        [b:GB62040] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62041] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62042] = 0;    // 運転中確認フラグのクリア
  }
}

// 4軸目
if([b:GB62050] == 1){
  [w:GD61000] = 4;

  // アラームチェック
  if([1-4:s32:400129] == 0){    // アラームが発生していない場合
    [b:GB62051] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62051] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-4:s32:400201] == 0){    // 運転していない場合
    [b:GB62052] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62052] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62051] == 1 && [b:GB62052] == 1){
    if([b:GB62055] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
        [b:GB62050] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62053] = 1;    // 運転準備開始トリガ
        [b:GB62051] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62052] = 0;    // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
        [b:GB62050] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62056] = 1;    // 運転終了開始トリガ
        [b:GB62051] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62052] = 0;    // 運転中確認フラグのクリア
    }
  } else {
        [b:GB62050] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62051] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62052] = 0;    // 運転中確認フラグのクリア
  }
```

```
}
```

スクリプト No.	31017	スクリプト名	Script31017
コメント	1-4_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62023

```
// タッチパネルから運転できるように準備するスクリプト
// NET-IN などの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 1-4 軸マルチ運転の画面スクリプト

// 1-4_マルチ運転準備スクリプト No.31017
// 1-4_マルチ運転開始_終了前確認スクリプト(No.31016)の GB62023、GB62033、GB62043、GB62053 の ON 中で、スクリプトを起動

// [b:GB62023], [b:GB62033], [b:GB62043], [b:GB62053] : 運転準備開始トリガ(このスクリプトの起動トリガ)
// [b:GB62024], [b:GB62034], [b:GB62044], [b:GB62054] : 運転準備中フラグ
// [b:GB62025], [b:GB62035], [b:GB62045], [b:GB62055] : タッチパネル運転中フラグ
// [u32:GD61202], [u32:GD61302], [u32:GD61402], [u32:GD61502] : NET-IN0～15 のバックアップデバイス
// [u32:GD61234], [u32:GD61334], [u32:GD61434], [u32:GD61534] : IN0～7 のバックアップデバイス
// [s32:GD61250], [s32:GD61350], [s32:GD61450], [s32:GD61550] : モニタの指令位置のバックアップデバイス
// [s32:GD61252], [s32:GD61352], [s32:GD61452], [u32:GD61552] : プリセット位置のバックアップデバイス
// [u32:GD61254], [u32:GD61354], [u32:GD61454], [u32:GD61554] : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD61256], [w:GD61356], [w:GD61456], [w:GD61556] : Configuration 実行待ち時間用タイマー

// 1 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62023] == 1 && [b:GB62024] == 0){

  [b:GB62024] = 1;                // 運転準備中フラグ(1:運転準備中)
  bmov([1-1:u32:404449],[u32:GD61202],16);  // NET-IN0～15 のバックアップ
  bmov([1-1:u32:404353],[u32:GD61234],8); // IN0～7 のバックアップ
  [s32:GD61252] = [1-1:s32:400909];       // プリセット位置のバックアップ
  [u32:GD61254] = [1-1:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-1:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-1:u32:404449] = 48; // NET-IN0  M0
  [1-1:u32:404451] = 49; // NET-IN1  M1
  [1-1:u32:404453] = 50; // NET-IN2  M2
  [1-1:u32:404455] = 51; // NET-IN3  M3
  [1-1:u32:404457] = 52; // NET-IN4  M4
  [1-1:u32:404459] = 53; // NET-IN5  M5
  [1-1:u32:404461] = 4;  // NET-IN6  START
  [1-1:u32:404463] = 3;  // NET-IN7  HOME
  [1-1:u32:404465] = 18; // NET-IN8  STOP
  [1-1:u32:404467] = 9;  // NET-IN9  MS1
  [1-1:u32:404469] = 10; // NET-IN10 MS2
  [1-1:u32:404471] = 5;  // NET-IN11 SSTART
  [1-1:u32:404473] = 6;  // NET-IN12 +JOG
  [1-1:u32:404475] = 7;  // NET-IN13 -JOG
```

```
  [1-1:u32:404477] = 1; // NET-IN14 FWD
  [1-1:u32:404479] = 2; // NET-IN15 RVS

  [1-1:u32:404353] = 32;   // IN0 R0
  [1-1:u32:404355] = 33;   // IN1 R1
  [1-1:u32:404357] = 34;   // IN2 R2
  [1-1:u32:404359] = 35;   // IN3 R3
  [1-1:u32:404361] = 36;   // IN4 R4
  [1-1:u32:404363] = 37;   // IN5 R5
  [1-1:u32:404365] = 18;   // IN6 STOP
  [1-1:u32:404367] = 39;   // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61250] = [1-1:s32:400199];
  [1-1:s32:400909] = [s32:GD61250];

  // Configuration の実行
  [1-1:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61256] > 1 && [b:GB62024] == 1){

  [1-1:s32:400395] = 1;         // P-PRESET(プリセット)実行する
  [1-1:s32:400909] = [s32:GD61252]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-1:s32:400397] = 0;         // Configuration 実行のゼロクリア
  [1-1:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62023] = 0;            // 運転準開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62024] = 0;            // 運転準備中フラグのクリア
  [b:GB62025] = 1;            // タッチパネル運転中フラグ(1:運転中)
}

// 2 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62033] == 1 && [b:GB62034] == 0){

  [b:GB62034] = 1;                  // 運転準備中フラグ(1:運転準備中)
  bmov([1-2:u32:404449],[u32:GD61302],16);  // NET-IN0～15 のバックアップ
  bmov([1-2:u32:404353],[u32:GD61334],8); // IN0～7 のバックアップ
  [s32:GD61352] = [1-2:s32:400909];       // プリセット位置のバックアップ
  [u32:GD61354] = [1-2:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-2:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-2:u32:404449] = 48; // NET-IN0  M0
  [1-2:u32:404451] = 49; // NET-IN1  M1
  [1-2:u32:404453] = 50; // NET-IN2  M2
  [1-2:u32:404455] = 51; // NET-IN3  M3
  [1-2:u32:404457] = 52; // NET-IN4  M4
```

```
  [1-2:u32:404459] = 53; // NET-IN5  M5
  [1-2:u32:404461] = 4;  // NET-IN6  START
  [1-2:u32:404463] = 3;  // NET-IN7  HOME
  [1-2:u32:404465] = 18; // NET-IN8  STOP
  [1-2:u32:404467] = 9;  // NET-IN9  MS1
  [1-2:u32:404469] = 10; // NET-IN10 MS2
  [1-2:u32:404471] = 5;  // NET-IN11 SSTART
  [1-2:u32:404473] = 6;  // NET-IN12 +JOG
  [1-2:u32:404475] = 7;  // NET-IN13 -JOG
  [1-2:u32:404477] = 1;  // NET-IN14 FWD
  [1-2:u32:404479] = 2;  // NET-IN15 RVS

  [1-2:u32:404353] = 32;   // IN0 R0
  [1-2:u32:404355] = 33;   // IN1 R1
  [1-2:u32:404357] = 34;   // IN2 R2
  [1-2:u32:404359] = 35;   // IN3 R3
  [1-2:u32:404361] = 36;   // IN4 R4
  [1-2:u32:404363] = 37;   // IN5 R5
  [1-2:u32:404365] = 18;   // IN6 STOP
  [1-2:u32:404367] = 39;   // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61350] = [1-2:s32:400199];
  [1-2:s32:400909] = [s32:GD61350];

  // Configuration の実行
  [1-2:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61356] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61356] > 1 && [b:GB62034] == 1){

  [1-2:s32:400395] = 1;         // P-PRESET(プリセット)実行する
  [1-2:s32:400909] = [s32:GD61352];  // バックアップしたプリセット位置パラメータを元に戻す
  [1-2:s32:400397] = 0;         // Configuration 実行のゼロクリア
  [1-2:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62033] = 0;            // 運転準備開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
  [b:GB62034] = 0;            // 運転準備中フラグのクリア
  [b:GB62035] = 1;            // タッチパネル運転中フラグ(1:運転中)
}

// 3 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62043] == 1 && [b:GB62044] == 0){

  [b:GB62044] = 1;                  // 運転準備中フラグ(1:運転準備中)
  bmov([1-3:u32:404449],[u32:GD61402],16);  // NET-IN0～15 のバックアップ
  bmov([1-3:u32:404353],[u32:GD61434],8); // IN0～7 のバックアップ
  [s32:GD61452] = [1-3:s32:400909];       // プリセット位置のバックアップ
  [u32:GD61454] = [1-3:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)
```

```
  // NET-INの入力値をクリアする
  [1-3:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-3:u32:404449] = 48;// NET-IN0  M0
  [1-3:u32:404451] = 49;// NET-IN1  M1
  [1-3:u32:404453] = 50;// NET-IN2  M2
  [1-3:u32:404455] = 51;// NET-IN3  M3
  [1-3:u32:404457] = 52;// NET-IN4  M4
  [1-3:u32:404459] = 53;// NET-IN5  M5
  [1-3:u32:404461] = 4; // NET-IN6  START
  [1-3:u32:404463] = 3; // NET-IN7  HOME
  [1-3:u32:404465] = 18;// NET-IN8  STOP
  [1-3:u32:404467] = 9; // NET-IN9  MS1
  [1-3:u32:404469] = 10;// NET-IN10 MS2
  [1-3:u32:404471] = 5; // NET-IN11 SSTART
  [1-3:u32:404473] = 6; // NET-IN12 +JOG
  [1-3:u32:404475] = 7; // NET-IN13 -JOG
  [1-3:u32:404477] = 1; // NET-IN14 FWD
  [1-3:u32:404479] = 2; // NET-IN15 RVS

  [1-3:u32:404353] = 32;   // IN0 R0
  [1-3:u32:404355] = 33;   // IN1 R1
  [1-3:u32:404357] = 34;   // IN2 R2
  [1-3:u32:404359] = 35;   // IN3 R3
  [1-3:u32:404361] = 36;   // IN4 R4
  [1-3:u32:404363] = 37;   // IN5 R5
  [1-3:u32:404365] = 18;   // IN6 STOP
  [1-3:u32:404367] = 39;   // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61450] = [1-3:s32:400199];
  [1-3:s32:400909] = [s32:GD61450];

  // Configurationの実行
  [1-3:s32:400397] = 1;

  // Configuration実行待ち時間のタイマー記録(単位は秒)
  [w:GD61456] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61456] > 1 && [b:GB62044] == 1){

  [1-3:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-3:s32:400909] = [s32:GD61452];  // バックアップしたプリセット位置パラメータを元に戻す
  [1-3:s32:400397] = 0;          // Configuration実行のゼロクリア
  [1-3:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62043] = 0;            // 運転準備開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
  [b:GB62044] = 0;            // 運転準備中フラグのクリア
  [b:GB62045] = 1;            // タッチパネル運転中フラグ(1:運転中)
}

// 4軸目
```

```
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62053] == 1 && [b:GB62054] == 0){

   [b:GB62054] = 1;                  // 運転準備中フラグ(1:運転準備中)
   bmov([1-4:u32:404449],[u32:GD61502],16);  // NET-IN0～15 のバックアップ
   bmov([1-4:u32:404353],[u32:GD61534],8); // IN0～7 のバックアップ
   [s32:GD61552] = [1-4:s32:400909];       // プリセット位置のバックアップ
   [u32:GD61554] = [1-4:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

   // NET-IN の入力値をクリアする
   [1-4:w:400126] = 0;

   // タッチパネルから運転する為に信号の割当を変更する
   [1-4:u32:404449] = 48;// NET-IN0  M0
   [1-4:u32:404451] = 49;// NET-IN1  M1
   [1-4:u32:404453] = 50;// NET-IN2  M2
   [1-4:u32:404455] = 51;// NET-IN3  M3
   [1-4:u32:404457] = 52;// NET-IN4  M4
   [1-4:u32:404459] = 53;// NET-IN5  M5
   [1-4:u32:404461] = 4; // NET-IN6  START
   [1-4:u32:404463] = 3; // NET-IN7  HOME
   [1-4:u32:404465] = 18;// NET-IN8  STOP
   [1-4:u32:404467] = 9; // NET-IN9  MS1
   [1-4:u32:404469] = 10;// NET-IN10 MS2
   [1-4:u32:404471] = 5; // NET-IN11 SSTART
   [1-4:u32:404473] = 6; // NET-IN12 +JOG
   [1-4:u32:404475] = 7; // NET-IN13 -JOG
   [1-4:u32:404477] = 1; // NET-IN14 FWD
   [1-4:u32:404479] = 2; // NET-IN15 RVS

   [1-4:u32:404353] = 32;   // IN0 R0
   [1-4:u32:404355] = 33;   // IN1 R1
   [1-4:u32:404357] = 34;   // IN2 R2
   [1-4:u32:404359] = 35;   // IN3 R3
   [1-4:u32:404361] = 36;   // IN4 R4
   [1-4:u32:404363] = 37;   // IN5 R5
   [1-4:u32:404365] = 18;   // IN6 STOP
   [1-4:u32:404367] = 39;   // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD61550] = [1-4:s32:400199];
   [1-4:s32:400909] = [s32:GD61550];

   // Configuration の実行
   [1-4:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD61556] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61556] > 1 && [b:GB62054] == 1){

   [1-4:s32:400395] = 1;         // P-PRESET(プリセット)実行する
   [1-4:s32:400909] = [s32:GD61552];  // バックアップしたプリセット位置パラメータを元に戻す
   [1-4:s32:400397] = 0;         // Configuration 実行のゼロクリア
```

```
    [1-4:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62053] = 0;             // 運転準備開始トリガのクリア
                             // (このスクリプトの起動トリガを終了する)
    [b:GB62054] = 0;             // 運転準備中フラグのクリア
    [b:GB62055] = 1;             // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプト No.	31018	スクリプト名	Script31018
コメント	1-4_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62025

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データ No.を M0~M5(割付直した NET-IN)に反映させる
// 反映した M0~M5 は、ドライバ入力指令(400126)にて、ON する

// 1-4_マルチ運転中スクリプト No.31018
// 1-4_マルチ運転準備スクリプト(No.31017)の GB62025、GB62035、GB62045、GB62055 の ON 中で、スクリプトを起動

// [b:GB62025], [b:GB62035], [b:GB62045], [b:GB62055] : タッチパネル運転中フラグ(このスクリプトの起動トリガ)
// また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-1:u32:400195] ~ [1-4:u32:400195] : モニタの現在の選択データ No.の Modbus アドレス
// [u32:GD61200], [u32GD61300], [GD61400], [u32:GD61500] : 運転データ No.のデバイス

// 選択した運転データ No.をドライバ入力指令の M0~M5 に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データ No.(400195)と選択した運転データ No.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

// 1 軸目
if([b:GB62025] == 1){

    if([1-1:u32:400195] != [u32:GD61200]){
        [1-1:b:400126.b0] = [b:GD61200.b0];     // M0
        [1-1:b:400126.b1] = [b:GD61200.b1];     // M1
        [1-1:b:400126.b2] = [b:GD61200.b2];     // M2
        [1-1:b:400126.b3] = [b:GD61200.b3];     // M3
        [1-1:b:400126.b4] = [b:GD61200.b4];     // M4
        [1-1:b:400126.b5] = [b:GD61200.b5];     // M5
    }
}

// 2 軸目
if([b:GB62035] == 1){

    if([1-2:u32:400195] != [u32:GD61300]){
        [1-2:b:400126.b0] = [b:GD61300.b0];     // M0
        [1-2:b:400126.b1] = [b:GD61300.b1];     // M1
        [1-2:b:400126.b2] = [b:GD61300.b2];     // M2
        [1-2:b:400126.b3] = [b:GD61300.b3];     // M3
        [1-2:b:400126.b4] = [b:GD61300.b4];     // M4
        [1-2:b:400126.b5] = [b:GD61300.b5];     // M5
    }
}
```

```
// 3 軸目
if([b:GB62045] == 1){

   if([1-3:u32:400195] != [u32:GD61400]){
      [1-3:b:400126.b0] = [b:GD61400.b0];      // M0
      [1-3:b:400126.b1] = [b:GD61400.b1];      // M1
      [1-3:b:400126.b2] = [b:GD61400.b2];      // M2
      [1-3:b:400126.b3] = [b:GD61400.b3];      // M3
      [1-3:b:400126.b4] = [b:GD61400.b4];      // M4
      [1-3:b:400126.b5] = [b:GD61400.b5];      // M5
   }
}

// 4 軸目
if([b:GB62055] == 1){

   if([1-4:u32:400195] != [u32:GD61500]){
      [1-4:b:400126.b0] = [b:GD61500.b0];      // M0
      [1-4:b:400126.b1] = [b:GD61500.b1];      // M1
      [1-4:b:400126.b2] = [b:GD61500.b2];      // M2
      [1-4:b:400126.b3] = [b:GD61500.b3];      // M3
      [1-4:b:400126.b4] = [b:GD61500.b4];      // M4
      [1-4:b:400126.b5] = [b:GD61500.b5];      // M5
   }
}
```

スクリプト No.	31019	スクリプト名	Script31019
コメント	1-4_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62026

```
// タッチパネルからの運転を終了するスクリプト
// バックアップした NET-IN などの信号を元に戻したり、必要なパラメータなどを元に戻す
// 1-4 軸のマルチ運転の画面スクリプト

// 1-4_マルチ運転終了スクリプト No.31019
// 1-4_マルチ運転開始_終了前確認スクリプト(No.31016)の GB62026、GB62036、GB62046、GB62056 の ON 中
で、スクリプトを起動

// [b:GB62025],    [b:GB62035],    [b:GB62045],    [b:GB62055]    : タッチパネル運転中フラグ
// [b:GB62026],    [b:GB62036],    [b:GB62046],    [b:GB62056]    : 運転終了開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62027],    [b:GB62037],    [b:GB62047],    [b:GB62057]    : 運転終了中フラグ
// [u32:GD61202], [u32:GD61302], [u32:GD61402], [u32:GD61502] : NET-IN0～15 のバックアップデバイ
ス
// [u32:GD61234], [u32:GD61334], [u32:GD61434], [u32:GD61534]  : IN0～7 のバックアップデバイス
// [s32:GD61250], [s32:GD61350], [s32:GD61450], [s32:GD61550]  : モニタの指令位置のバックアップ
デバイス
// [s32:GD61252], [s32:GD61352], [s32:GD61452], [s32:GD61552]  : プリセット位置のバックアップデ
バイス
// [u32:GD61254], [u32:GD61354], [u32:GD61454], [u32:GD61554] : JOG 移動量のバックアップデバイス
(GOT の最小移動量に使用)
// [w:GD61256],   [w:GD61356],    [w:GD61456],    [w:GD61556]      : Configuration 実行待ち時間用
タイマー

// 1 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62026] == 1 && [b:GB62027] == 0){
```

```
    [b:GB62025] = 0;                  // タッチパネル運転中フラグのクリア
    [b:GB62027] = 1;                  // 運転終了中フラグ(1：運転終了中)
    bmov([u32:GD61202],[1-1:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
    bmov([u32:GD61234],[1-1:u32:404353],8); // バックアップした IN0～7 を元に戻す
    [1-1:u32:404169] = [u32:GD61254];       // バックアップした JOG 移動量を元に戻す

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD61250] = [1-1:s32:400199];
    [1-1:s32:400909] = [s32:GD61250];

    // NET-IN の入力値をクリアする
    [1-1:w:400126] = 0;

    // Configuration の実行
    [1-1:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD61256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61256] > 1 && [b:GB62027] == 1){

    [1-1:s32:400395] = 1;          // P-PRESET(プリセット)実行する
    [1-1:s32:400909] = [s32:GD61252];  // プリセット位置パラメータを元に戻す
    [1-1:s32:400397] = 0;          // Configuration 実行のゼロクリア
    [1-1:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62026] = 0;            // 運転終了開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
    [b:GB62027] = 0;            // 運転終了中フラグのクリア
}

// 2 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62036] == 1 && [b:GB62037] == 0){

    [b:GB62035] = 0;                  // タッチパネル運転中フラグのクリア
    [b:GB62037] = 1;                  // 運転終了中フラグ(1：運転終了中)
    bmov([u32:GD61302],[1-2:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
    bmov([u32:GD61334],[1-2:u32:404353],8); // バックアップした IN0～7 を元に戻す
    [1-2:u32:404169] = [u32:GD61354];       // バックアップした JOG 移動量を元に戻す

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD61350] = [1-2:s32:400199];
    [1-2:s32:400909] = [s32:GD61350];

    // NET-IN の入力値をクリアする
    [1-2:w:400126] = 0;

    // Configuration の実行
    [1-2:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD61356] = [w:GS7];
}
```

```
// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61356] > 1 && [b:GB62037] == 1){

  [1-2:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-2:s32:400909] = [s32:GD61352];  // プリセット位置パラメータを元に戻す
  [1-2:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-2:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62036] = 0;            // 運転終了開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62037] = 0;            // 運転終了中フラグのクリア
}

// 3 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62046] == 1 && [b:GB62047] == 0){

  [b:GB62045] = 0;                // タッチパネル運転中フラグのクリア
  [b:GB62047] = 1;                // 運転終了中フラグ(1 : 運転終了中)
  bmov([u32:GD61402],[1-3:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD61434],[1-3:u32:404353],8); // バックアップした IN0～7 を元に戻す
  [1-3:u32:404169] = [u32:GD61454];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61450] = [1-3:s32:400199];
  [1-3:s32:400909] = [s32:GD61450];

  // NET-IN の入力値をクリアする
  [1-3:w:400126] = 0;

  // Configuration の実行
  [1-3:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61456] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61456] > 1 && [b:GB62047] == 1){

  [1-3:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-3:s32:400909] = [s32:GD61452];  // プリセット位置パラメータを元に戻す
  [1-3:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-3:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62046] = 0;            // 運転終了開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62047] = 0;            // 運転終了中フラグのクリア
}

// 4 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62056] == 1 && [b:GB62057] == 0){

  [b:GB62055] = 0;                // タッチパネル運転中フラグのクリア
```

```
    [b:GB62057] = 1;                // 運転終了中フラグ(1：運転終了中)
    bmov([u32:GD61502],[1-4:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
    bmov([u32:GD61534],[1-4:u32:404353],8); // バックアップした IN0～7 を元に戻す
    [1-4:u32:404169] = [u32:GD61554];       // バックアップした JOG 移動量を元に戻す

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD61550] = [1-4:s32:400199];
    [1-4:s32:400909] = [s32:GD61550];

    // NET-IN の入力値をクリアする
    [1-4:w:400126] = 0;

    // Configuration の実行
    [1-4:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD61556] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61556] > 1 && [b:GB62057] == 1){

    [1-4:s32:400395] = 1;         // P-PRESET(プリセット)実行する
    [1-4:s32:400909] = [s32:GD61552];  // プリセット位置パラメータを元に戻す
    [1-4:s32:400397] = 0;         // Configuration 実行のゼロクリア
    [1-4:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62056] = 0;           // 運転終了開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
    [b:GB62057] = 0;           // 運転終了中フラグのクリア
}
```

スクリプト No.	31016	スクリプト名	Script31016
コメント	1-4_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) B62030

ベース画面 31022 のスクリプト No.31016 と同じです。

スクリプト No.	31017	スクリプト名	Script31017
コメント	1-4_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62033

ベース画面 31022 のスクリプト No.31017 同じです。

スクリプト No.	31018	スクリプト名	Script31018
コメント	1-4_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62035

ベース画面 31022 のスクリプト No.31018 と同じです。

スクリプト No.	31019	スクリプト名	Script31019
コメント	1-4_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62036

ベース画面 31022 のスクリプト No.31019 と同じです。

スクリプト No.	31016	スクリプト名	Script31016
コメント	1-4_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62040

ベース画面31022のスクリプトNo.31016と同じです。			
スクリプトNo.	31017	スクリプト名	Script31017
コメント	1-4_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62043
ベース画面31022のスクリプトNo.31017同じです。			
スクリプトNo.	31018	スクリプト名	Script31018
コメント	1-4_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62045
ベース画面31022のスクリプトNo.31018と同じです。			
スクリプトNo.	31019	スクリプト名	Script31019
コメント	1-4_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62046
ベース画面31022のスクリプトNo.31019と同じです。			
スクリプトNo.	31016	スクリプト名	Script31016
コメント	1-4_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62050
ベース画面31022のスクリプトNo.31016と同じです。			
スクリプトNo.	31017	スクリプト名	Script31017
コメント	1-4_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62053
ベース画面31022のスクリプトNo.31017同じです。			
スクリプトNo.	31018	スクリプト名	Script31018
コメント	1-4_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62055
ベース画面31022のスクリプトNo.31018と同じです。			
スクリプトNo.	31019	スクリプト名	Script31019
コメント	1-4_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62056
ベース画面31022のスクリプトNo.31019と同じです。			

ベース画面31023

スクリプトNo.	31020	スクリプト名	Script31020
コメント	5-8_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62060

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 5-8軸のマルチ運転の画面スクリプト

// 5-8_マルチ運転開始_終了前確認スクリプト No.31020
// 5-8軸のマルチ運転画面のそれぞれの開始ボタンのON中周期、2秒で、スクリプトを起動

// [b:GB62060] : 5軸目の起動トリガ、ビットセットで起動
// [b:GB62070] : 6軸目の起動トリガ、ビットセットで起動
// [b:GB62080] : 7軸目の起動トリガ、ビットセットで起動
// [b:GB62090] : 8軸目の起動トリガ、ビットセットで起動
```

```
// [b:GB62065] : 5軸目のタッチパネル運転中フラグ
// [b:GB62075] : 6軸目のタッチパネル運転中フラグ
// [b:GB62085] : 7軸目のタッチパネル運転中フラグ
// [b:GB62095] : 8軸目のタッチパネル運転中フラグ

// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 5軸目
if([b:GB62060] == 1){
  [w:GD61000] = 5;

  // アラームチェック
  if([1-5:s32:400129] == 0){    // アラームが発生していない場合
    [b:GB62061] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62061] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-5:s32:400201] == 0){    // 運転していない場合
    [b:GB62062] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62062] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62061] == 1 && [b:GB62062] == 1){
    if([b:GB62065] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62060] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62063] = 1;    // 運転準備開始トリガ
      [b:GB62061] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62062] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62060] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62066] = 1;    // 運転終了開始トリガ
      [b:GB62061] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62062] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62060] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62061] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62062] = 0;    // 運転中確認フラグのクリア
  }
}

// 6軸目
if([b:GB62070] == 1){
  [w:GD61000] = 6;

  // アラームチェック
```

```
    if([1-6:s32:400129] == 0){     // アラームが発生していない場合
      [b:GB62071] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
      [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
    } else {                  // アラームが発生している場合
      [b:GB62071] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
      [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
    }

    // 運転中チェック
    if([1-6:s32:400201] == 0){     // 運転していない場合
      [b:GB62072] = 1;        // 運転中確認フラグ(1:運転していない)
    } else {                  // 運転中の場合
      [b:GB62072] = 0;        // 運転中確認フラグ(0:運転中)
      [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
    }

    // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
    if([b:GB62071] == 1 && [b:GB62072] == 1){
      if([b:GB62075] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
        [b:GB62070] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62073] = 1;    // 運転準備開始トリガ
        [b:GB62071] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62072] = 0;    // 運転中確認フラグのクリア
      } else {                // タッチパネル運転中なら、運転終了準備をする
        [b:GB62070] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62076] = 1;    // 運転終了開始トリガ
        [b:GB62071] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62072] = 0;    // 運転中確認フラグのクリア
      }
    } else {
        [b:GB62070] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62071] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62072] = 0;    // 運転中確認フラグのクリア
    }
}

// 7軸目
if([b:GB62080] == 1){
    [w:GD61000] = 7;

    // アラームチェック
    if([1-7:s32:400129] == 0){     // アラームが発生していない場合
      [b:GB62081] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
      [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
    } else {                  // アラームが発生している場合
      [b:GB62081] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
      [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
    }

    // 運転中チェック
    if([1-7:s32:400201] == 0){     // 運転していない場合
      [b:GB62082] = 1;        // 運転中確認フラグ(1:運転していない)
    } else {                  // 運転中の場合
      [b:GB62082] = 0;        // 運転中確認フラグ(0:運転中)
      [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
    }
```

```
  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62081] == 1 && [b:GB62082] == 1){
    if([b:GB62085] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62080] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62083] = 1;    // 運転準備開始トリガ
      [b:GB62081] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62082] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62080] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62086] = 1;    // 運転終了開始トリガ
      [b:GB62081] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62082] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62080] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62081] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62082] = 0;    // 運転中確認フラグのクリア
  }
}

// 8軸目
if([b:GB62090] == 1){
  [w:GD61000] = 8;

  // アラームチェック
  if([1-8:s32:400129] == 0){    // アラームが発生していない場合
    [b:GB62091] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62091] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-8:s32:400201] == 0){    // 運転していない場合
    [b:GB62092] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62092] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62091] == 1 && [b:GB62092] == 1){
    if([b:GB62095] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62090] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62093] = 1;    // 運転準備開始トリガ
      [b:GB62091] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62092] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62090] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62096] = 1;    // 運転終了開始トリガ
      [b:GB62091] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62092] = 0;    // 運転中確認フラグのクリア
    }
  } else {
```

```
        [b:GB62090] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62091] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62092] = 0;    // 運転中確認フラグのクリア
    }
}
```

スクリプトNo.	31021	スクリプト名	Script31021
コメント	5-8_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62063

```
// タッチパネルから運転できるように準備するスクリプト
// NET-INなどの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 5-8軸マルチ運転の画面スクリプト

// 5-8_マルチ運転準備スクリプト No.31021
// 5-8_マルチ運転開始_終了前確認スクリプト(No.31020)のGB62063、GB62073、GB62083、GB62093のON中で、スクリプトを起動

// [b:GB62063],    [b:GB62073],    [b:GB62083],    [b:GB62093]   : 運転準備開始トリガ(このスクリプトの起動トリガ)
// [b:GB62064],    [b:GB62074],    [b:GB62084],    [b:GB62094]   : 運転準備中フラグ
// [b:GB62065],    [b:GB62075],    [b:GB62085],    [b:GB62095]   : タッチパネル運転中フラグ
// [u32:GD61602], [u32:GD61702], [u32:GD61802], [u32:GD61902] : NET-IN0～15のバックアップデバイス
// [u32:GD61634], [u32:GD61734], [u32:GD61834], [u32:GD61934] : IN0～7のバックアップデバイス
// [s32:GD61650], [s32:GD61750], [s32:GD61850], [s32:GD61950] : モニタの指令位置のバックアップデバイス
// [s32:GD61652], [s32:GD61752], [s32:GD61852], [u32:GD61952] : プリセット位置のバックアップデバイス
// [u32:GD61654], [u32:GD61754], [u32:GD61854], [u32:GD61954] : JOG移動量のバックアップデバイス(GOTの最小移動量に使用)
// [w:GD61656],   [w:GD61756],    [w:GD61856],   [w:GD61956]     : Configuration 実行待ち時間用タイマー

// 5軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62063] == 1 && [b:GB62064] == 0){

    [b:GB62064] = 1;                    // 運転準備中フラグ(1:運転準備中)
    bmov([1-5:u32:404449],[u32:GD61602],16);  // NET-IN0～15のバックアップ
    bmov([1-5:u32:404353],[u32:GD61634],8); // IN0～7のバックアップ
    [s32:GD61652] = [1-5:s32:400909];       // プリセット位置のバックアップ
    [u32:GD61654] = [1-5:u32:404169];       // JOG移動量のバックアップ(GOTの最小移動量に使用)

    // NET-INの入力値をクリアする
    [1-5:w:400126] = 0;

    // タッチパネルから運転する為に信号の割当を変更する
    [1-5:u32:404449] = 48; // NET-IN0  M0
    [1-5:u32:404451] = 49; // NET-IN1  M1
    [1-5:u32:404453] = 50; // NET-IN2  M2
    [1-5:u32:404455] = 51; // NET-IN3  M3
    [1-5:u32:404457] = 52; // NET-IN4  M4
    [1-5:u32:404459] = 53; // NET-IN5  M5
    [1-5:u32:404461] = 4;  // NET-IN6  START
    [1-5:u32:404463] = 3;  // NET-IN7  HOME
    [1-5:u32:404465] = 18; // NET-IN8  STOP
    [1-5:u32:404467] = 9;  // NET-IN9  MS1
    [1-5:u32:404469] = 10; // NET-IN10 MS2
```

```
  [1-5:u32:404471] = 5; // NET-IN11 SSTART
  [1-5:u32:404473] = 6; // NET-IN12 +JOG
  [1-5:u32:404475] = 7; // NET-IN13 -JOG
  [1-5:u32:404477] = 1; // NET-IN14 FWD
  [1-5:u32:404479] = 2; // NET-IN15 RVS

  [1-5:u32:404353] = 32;   // IN0 R0
  [1-5:u32:404355] = 33;   // IN1 R1
  [1-5:u32:404357] = 34;   // IN2 R2
  [1-5:u32:404359] = 35;   // IN3 R3
  [1-5:u32:404361] = 36;   // IN4 R4
  [1-5:u32:404363] = 37;   // IN5 R5
  [1-5:u32:404365] = 18;   // IN6 STOP
  [1-5:u32:404367] = 39;   // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61650] = [1-5:s32:400199];
  [1-5:s32:400909] = [s32:GD61650];

  // Configuration の実行
  [1-5:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61656] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61656] > 1 && [b:GB62064] == 1){

  [1-5:s32:400395] = 1;         // P-PRESET(プリセット)実行する
  [1-5:s32:400909] = [s32:GD61652]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-5:s32:400397] = 0;         // Configuration 実行のゼロクリア
  [1-5:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62063] = 0;           // 運転準備開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62064] = 0;           // 運転準備中フラグのクリア
  [b:GB62065] = 1;           // タッチパネル運転中フラグ(1:運転中)
}

// 6 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62073] == 1 && [b:GB62074] == 0){

  [b:GB62074] = 1;                // 運転準備中フラグ(1:運転準備中)
  bmov([1-6:u32:404449],[u32:GD61702],16);  // NET-IN0～15 のバックアップ
  bmov([1-6:u32:404353],[u32:GD61734],8); // IN0～7 のバックアップ
  [s32:GD61752] = [1-6:s32:400909];      // プリセット位置のバックアップ
  [u32:GD61754] = [1-6:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-6:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-6:u32:404449] = 48;// NET-IN0  M0
  [1-6:u32:404451] = 49;// NET-IN1  M1
```

```
   [1-6:u32:404453] = 50; // NET-IN2  M2
   [1-6:u32:404455] = 51; // NET-IN3  M3
   [1-6:u32:404457] = 52; // NET-IN4  M4
   [1-6:u32:404459] = 53; // NET-IN5  M5
   [1-6:u32:404461] = 4;  // NET-IN6  START
   [1-6:u32:404463] = 3;  // NET-IN7  HOME
   [1-6:u32:404465] = 18; // NET-IN8  STOP
   [1-6:u32:404467] = 9;  // NET-IN9  MS1
   [1-6:u32:404469] = 10; // NET-IN10 MS2
   [1-6:u32:404471] = 5;  // NET-IN11 SSTART
   [1-6:u32:404473] = 6;  // NET-IN12 +JOG
   [1-6:u32:404475] = 7;  // NET-IN13 -JOG
   [1-6:u32:404477] = 1;  // NET-IN14 FWD
   [1-6:u32:404479] = 2;  // NET-IN15 RVS

   [1-6:u32:404353] = 32;   // IN0 R0
   [1-6:u32:404355] = 33;   // IN1 R1
   [1-6:u32:404357] = 34;   // IN2 R2
   [1-6:u32:404359] = 35;   // IN3 R3
   [1-6:u32:404361] = 36;   // IN4 R4
   [1-6:u32:404363] = 37;   // IN5 R5
   [1-6:u32:404365] = 18;   // IN6 STOP
   [1-6:u32:404367] = 39;   // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD61750] = [1-6:s32:400199];
   [1-6:s32:400909] = [s32:GD61750];

   // Configuration の実行
   [1-6:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD61756] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61756] > 1 && [b:GB62074] == 1){

   [1-6:s32:400395] = 1;          // P-PRESET(プリセット)実行する
   [1-6:s32:400909] = [s32:GD61752];  // バックアップしたプリセット位置パラメータを元に戻す
   [1-6:s32:400397] = 0;          // Configuration 実行のゼロクリア
   [1-6:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62073] = 0;            // 運転準備開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
   [b:GB62074] = 0;            // 運転準備中フラグのクリア
   [b:GB62075] = 1;            // タッチパネル運転中フラグ(1:運転中)
}

// 7 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62083] == 1 && [b:GB62084] == 0){

   [b:GB62084] = 1;                  // 運転準備中フラグ(1:運転準備中)
   bmov([1-7:u32:404449],[u32:GD61802],16);  // NET-IN0～15 のバックアップ
   bmov([1-7:u32:404353],[u32:GD61834],8); // IN0～7 のバックアップ
```

```
  [s32:GD61852] = [1-7:s32:400909];      // プリセット位置のバックアップ
  [u32:GD61854] = [1-7:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-7:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-7:u32:404449] = 48; // NET-IN0  M0
  [1-7:u32:404451] = 49; // NET-IN1  M1
  [1-7:u32:404453] = 50; // NET-IN2  M2
  [1-7:u32:404455] = 51; // NET-IN3  M3
  [1-7:u32:404457] = 52; // NET-IN4  M4
  [1-7:u32:404459] = 53; // NET-IN5  M5
  [1-7:u32:404461] = 4;  // NET-IN6  START
  [1-7:u32:404463] = 3;  // NET-IN7  HOME
  [1-7:u32:404465] = 18; // NET-IN8  STOP
  [1-7:u32:404467] = 9;  // NET-IN9  MS1
  [1-7:u32:404469] = 10; // NET-IN10 MS2
  [1-7:u32:404471] = 5;  // NET-IN11 SSTART
  [1-7:u32:404473] = 6;  // NET-IN12 +JOG
  [1-7:u32:404475] = 7;  // NET-IN13 -JOG
  [1-7:u32:404477] = 1;  // NET-IN14 FWD
  [1-7:u32:404479] = 2;  // NET-IN15 RVS

  [1-7:u32:404353] = 32;   // IN0 R0
  [1-7:u32:404355] = 33;   // IN1 R1
  [1-7:u32:404357] = 34;   // IN2 R2
  [1-7:u32:404359] = 35;   // IN3 R3
  [1-7:u32:404361] = 36;   // IN4 R4
  [1-7:u32:404363] = 37;   // IN5 R5
  [1-7:u32:404365] = 18;   // IN6 STOP
  [1-7:u32:404367] = 39;   // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61850] = [1-7:s32:400199];
  [1-7:s32:400909] = [s32:GD61850];

  // Configuration の実行
  [1-7:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61856] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61856] > 1 && [b:GB62084] == 1){

  [1-7:s32:400395] = 1;        // P-PRESET(プリセット)実行する
  [1-7:s32:400909] = [s32:GD61852];  // バックアップしたプリセット位置パラメータを元に戻す
  [1-7:s32:400397] = 0;        // Configuration 実行のゼロクリア
  [1-7:s32:400395] = 0;        // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62083] = 0;           // 運転準備開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62084] = 0;           // 運転準備中フラグのクリア
  [b:GB62085] = 1;           // タッチパネル運転中フラグ(1:運転中)
```

```
}

// 8 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62093] == 1 && [b:GB62094] == 0){

  [b:GB62094] = 1;                   // 運転準備中フラグ(1：運転準備中)
  bmov([1-8:u32:404449],[u32:GD61902],16);  // NET-IN0～15 のバックアップ
  bmov([1-8:u32:404353],[u32:GD61934],8); // IN0～7 のバックアップ
  [s32:GD61952] = [1-8:s32:400909];       // プリセット位置のバックアップ
  [u32:GD61954] = [1-8:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-8:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-8:u32:404449] = 48;// NET-IN0  M0
  [1-8:u32:404451] = 49;// NET-IN1  M1
  [1-8:u32:404453] = 50;// NET-IN2  M2
  [1-8:u32:404455] = 51;// NET-IN3  M3
  [1-8:u32:404457] = 52;// NET-IN4  M4
  [1-8:u32:404459] = 53;// NET-IN5  M5
  [1-8:u32:404461] = 4; // NET-IN6  START
  [1-8:u32:404463] = 3; // NET-IN7  HOME
  [1-8:u32:404465] = 18;// NET-IN8  STOP
  [1-8:u32:404467] = 9; // NET-IN9  MS1
  [1-8:u32:404469] = 10;// NET-IN10 MS2
  [1-8:u32:404471] = 5; // NET-IN11 SSTART
  [1-8:u32:404473] = 6; // NET-IN12 +JOG
  [1-8:u32:404475] = 7; // NET-IN13 -JOG
  [1-8:u32:404477] = 1; // NET-IN14 FWD
  [1-8:u32:404479] = 2; // NET-IN15 RVS

  [1-8:u32:404353] = 32;   // IN0 R0
  [1-8:u32:404355] = 33;   // IN1 R1
  [1-8:u32:404357] = 34;   // IN2 R2
  [1-8:u32:404359] = 35;   // IN3 R3
  [1-8:u32:404361] = 36;   // IN4 R4
  [1-8:u32:404363] = 37;   // IN5 R5
  [1-8:u32:404365] = 18;   // IN6 STOP
  [1-8:u32:404367] = 39;   // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61950] = [1-8:s32:400199];
  [1-8:s32:400909] = [s32:GD61950];

  // Configuration の実行
  [1-8:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61956] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD61956] > 1 && [b:GB62094] == 1){
```

```
  [1-8:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-8:s32:400909] = [s32:GD61952];  // バックアップしたプリセット位置パラメータを元に戻す
  [1-8:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-8:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62093] = 0;            // 運転準備開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
  [b:GB62094] = 0;            // 運転準備中フラグのクリア
  [b:GB62095] = 1;            // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプト No.	31022	スクリプト名	Script31022
コメント	5-8_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62065

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データ No.を M0～M5(割付直した NET-IN)に反映させる
// 反映した M0～M5 は、ドライバ入力指令(400126)にて、ON する

// 5-8_マルチ運転中スクリプト No.31022
// 5-8_マルチ運転準備スクリプト(No.31021)の GB62065、GB62075、GB62085、GB62095 の ON 中で、スクリプト
を起動

// [b:GB62065], [b:GB62075], [b:GB62085], [b:GB62095] : タッチパネル運転中フラグ(このスクリプトの
起動トリガ)
// また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-5:u32:400195] ～ [1-8:u32:400195] : モニタの現在の選択データ No.の Modbus アドレス
// [u32:GD61600], [u32GD61700], [GD61800], [u32:GD61900] : 運転データ No.のデバイス

// 選択した運転データ No.をドライバ入力指令の M0～M5 に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データ No.(400195)と選択した運転データ No.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

// 5 軸目
if([b:GB62065] == 1){

  if([1-5:u32:400195] != [u32:GD61600]){
    [1-5:b:400126.b0] = [b:GD61600.b0];     // M0
    [1-5:b:400126.b1] = [b:GD61600.b1];     // M1
    [1-5:b:400126.b2] = [b:GD61600.b2];     // M2
    [1-5:b:400126.b3] = [b:GD61600.b3];     // M3
    [1-5:b:400126.b4] = [b:GD61600.b4];     // M4
    [1-5:b:400126.b5] = [b:GD61600.b5];     // M5
  }
}

// 6 軸目
if([b:GB62075] == 1){

  if([1-6:u32:400195] != [u32:GD61700]){
    [1-6:b:400126.b0] = [b:GD61700.b0];     // M0
    [1-6:b:400126.b1] = [b:GD61700.b1];     // M1
    [1-6:b:400126.b2] = [b:GD61700.b2];     // M2
    [1-6:b:400126.b3] = [b:GD61700.b3];     // M3
    [1-6:b:400126.b4] = [b:GD61700.b4];     // M4
    [1-6:b:400126.b5] = [b:GD61700.b5];     // M5
```

```
   }
}

// 7 軸目
if([b:GB62085] == 1){

   if([1-7:u32:400195] != [u32:GD61800]){
      [1-7:b:400126.b0] = [b:GD61800.b0];     // M0
      [1-7:b:400126.b1] = [b:GD61800.b1];     // M1
      [1-7:b:400126.b2] = [b:GD61800.b2];     // M2
      [1-7:b:400126.b3] = [b:GD61800.b3];     // M3
      [1-7:b:400126.b4] = [b:GD61800.b4];     // M4
      [1-7:b:400126.b5] = [b:GD61800.b5];     // M5
   }
}

// 8 軸目
if([b:GB62095] == 1){

   if([1-8:u32:400195] != [u32:GD61900]){
      [1-8:b:400126.b0] = [b:GD61900.b0];     // M0
      [1-8:b:400126.b1] = [b:GD61900.b1];     // M1
      [1-8:b:400126.b2] = [b:GD61900.b2];     // M2
      [1-8:b:400126.b3] = [b:GD61900.b3];     // M3
      [1-8:b:400126.b4] = [b:GD61900.b4];     // M4
      [1-8:b:400126.b5] = [b:GD61900.b5];     // M5
   }
}
```

スクリプト No.	31023	スクリプト名	Script31023
コメント	5-8_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62066

```
// タッチパネルからの運転を終了するスクリプト
// バックアップした NET-IN などの信号を元に戻したり、必要なパラメータなどを元に戻す
// 5-8 軸のマルチ運転の画面スクリプト

// 5-8_マルチ運転終了スクリプト No.31023
// 5-8_マルチ運転開始_終了前確認スクリプト(No.31020)の GB62066、GB62076、GB62086、GB62096 の ON 中で、スクリプトを起動

// [b:GB62065],    [b:GB62075],    [b:GB62085],    [b:GB62095]    : タッチパネル運転中フラグ
// [b:GB62066],    [b:GB62076],    [b:GB62086],    [b:GB62096]    : 運転終了開始トリガ(このスクリプトの起動トリガ)
// [b:GB62067],    [b:GB62077],    [b:GB62087],    [b:GB62097]    : 運転終了中フラグ
// [u32:GD61602], [u32:GD61702], [u32:GD61802], [u32:GD61902] : NET-IN0～15 のバックアップデバイス
// [u32:GD61634], [u32:GD61734], [u32:GD61834], [u32:GD61934] : IN0～7 のバックアップデバイス
// [s32:GD61650], [s32:GD61750], [s32:GD61850], [s32:GD61950] : モニタの指令位置のバックアップデバイス
// [s32:GD61652], [s32:GD61752], [s32:GD61852], [s32:GD61952] : プリセット位置のバックアップデバイス
// [u32:GD61654], [u32:GD61754], [u32:GD61854], [u32:GD61954] : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD61656],   [w:GD61756],    [w:GD61856],   [w:GD61956]     : Configuration 実行待ち時間用タイマー

// 5 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
```

```
if([b:GB62066] == 1 && [b:GB62067] == 0){

  [b:GB62065] = 0;                  // タッチパネル運転中フラグのクリア
  [b:GB62067] = 1;                  // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD61602],[1-5:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD61634],[1-5:u32:404353],8); // バックアップした IN0～7 を元に戻す
  [1-5:u32:404169] = [u32:GD61654];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61650] = [1-5:s32:400199];
  [1-5:s32:400909] = [s32:GD61650];

  // NET-IN の入力値をクリアする
  [1-5:w:400126] = 0;

  // Configuration の実行
  [1-5:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61656] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61656] > 1 && [b:GB62067] == 1){

  [1-5:s32:400395] = 1;         // P-PRESET(プリセット)実行する
  [1-5:s32:400909] = [s32:GD61652];  // プリセット位置パラメータを元に戻す
  [1-5:s32:400397] = 0;         // Configuration 実行のゼロクリア
  [1-5:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62066] = 0;            // 運転終了開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62067] = 0;            // 運転終了中フラグのクリア
}

// 6 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62076] == 1 && [b:GB62077] == 0){

  [b:GB62075] = 0;                  // タッチパネル運転中フラグのクリア
  [b:GB62077] = 1;                  // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD61702],[1-6:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD61734],[1-6:u32:404353],8); // バックアップした IN0～7 を元に戻す
  [1-6:u32:404169] = [u32:GD61754];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61750] = [1-6:s32:400199];
  [1-6:s32:400909] = [s32:GD61750];

  // NET-IN の入力値をクリアする
  [1-6:w:400126] = 0;

  // Configuration の実行
  [1-6:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
```

```
  [w:GD61756] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61756] > 1 && [b:GB62077] == 1){

  [1-6:s32:400395] = 1;         // P-PRESET(プリセット)実行する
  [1-6:s32:400909] = [s32:GD61752];  // プリセット位置パラメータを元に戻す
  [1-6:s32:400397] = 0;         // Configuration 実行のゼロクリア
  [1-6:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62076] = 0;            // 運転終了開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62077] = 0;            // 運転終了中フラグのクリア
}

// 7 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62086] == 1 && [b:GB62087] == 0){

  [b:GB62085] = 0;                  // タッチパネル運転中フラグのクリア
  [b:GB62087] = 1;                  // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD61802],[1-7:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD61834],[1-7:u32:404353],8); // バックアップした IN0～7 を元に戻す
  [1-7:u32:404169] = [u32:GD61854];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61850] = [1-7:s32:400199];
  [1-7:s32:400909] = [s32:GD61850];

  // NET-IN の入力値をクリアする
  [1-7:w:400126] = 0;

  // Configuration の実行
  [1-7:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61856] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61856] > 1 && [b:GB62087] == 1){

  [1-7:s32:400395] = 1;         // P-PRESET(プリセット)実行する
  [1-7:s32:400909] = [s32:GD61852];  // プリセット位置パラメータを元に戻す
  [1-7:s32:400397] = 0;         // Configuration 実行のゼロクリア
  [1-7:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62086] = 0;            // 運転終了開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62087] = 0;            // 運転終了中フラグのクリア
}

// 8 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62096] == 1 && [b:GB62097] == 0){
```

```
  [b:GB62095] = 0;                // タッチパネル運転中フラグのクリア
  [b:GB62097] = 1;                // 運転終了中フラグ(1：運転終了中)
  bmov([u32:GD61902],[1-8:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD61934],[1-8:u32:404353],8); // バックアップした IN0～7 を元に戻す
  [1-8:u32:404169] = [u32:GD61954];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD61950] = [1-8:s32:400199];
  [1-8:s32:400909] = [s32:GD61950];

  // NET-IN の入力値をクリアする
  [1-8:w:400126] = 0;

  // Configuration の実行
  [1-8:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD61956] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD61956] > 1 && [b:GB62097] == 1){

  [1-8:s32:400395] = 1;         // P-PRESET(プリセット)実行する
  [1-8:s32:400909] = [s32:GD61952];  // プリセット位置パラメータを元に戻す
  [1-8:s32:400397] = 0;         // Configuration 実行のゼロクリア
  [1-8:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62096] = 0;            // 運転終了開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62097] = 0;            // 運転終了中フラグのクリア
}
```

スクリプト No.	31020	スクリプト名	Script31020
コメント	5-8_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62070
ベース画面 31023 のスクリプト No.31020 と同じです。			

スクリプト No.	31021	スクリプト名	Script31021
コメント	5-8_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62073
ベース画面 31023 のスクリプト No.31021 と同じです。			

スクリプト No.	31022	スクリプト名	Script31022
コメント	5-8_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62075
ベース画面 31023 のスクリプト No.31022 と同じです。			

スクリプト No.	31023	スクリプト名	Script31023
コメント	5-8_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62076
ベース画面 31023 のスクリプト No.31023 と同じです。			

スクリプト No.	31020	スクリプト名	Script31020

コメント	5-8_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62080
ベース画面 31023 のスクリプト No.31020 と同じです。			

スクリプト No.	31021	スクリプト名	Script31021
コメント	5-8_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62083
ベース画面 31023 のスクリプト No.31021 と同じです。			

スクリプト No.	31022	スクリプト名	Script31022
コメント	5-8_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62085
ベース画面 31023 のスクリプト No.31022 と同じです。			

スクリプト No.	31023	スクリプト名	Script31023
コメント	5-8_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62086
ベース画面 31023 のスクリプト No.31023 と同じです。			

スクリプト No.	31020	スクリプト名	Script31020
コメント	5-8_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62090
ベース画面 31023 のスクリプト No.31020 と同じです。			

スクリプト No.	31021	スクリプト名	Script31021
コメント	5-8_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62093
ベース画面 31023 のスクリプト No.31021 と同じです。			

スクリプト No.	31022	スクリプト名	Script31022
コメント	5-8_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62095
ベース画面 31023 のスクリプト No.31022 と同じです。			

スクリプト No.	31023	スクリプト名	Script31023
コメント	5-8_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62096
ベース画面 31023 のスクリプト No.31023 と同じです。			

ベース画面 31024

スクリプト No.	31024	スクリプト名	Script31024
コメント	9-12_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62100

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 9-12 軸のマルチ運転の画面スクリプト

// 9-12_マルチ運転開始_終了前確認スクリプト No.31024
// 9-12 軸のマルチ運転画面のそれぞれの開始ボタンの ON 中周期、2 秒で、スクリプトを起動

// [b:GB62100] : 9 軸目の起動トリガ、ビットセットで起動
// [b:GB62110] : 10 軸目の起動トリガ、ビットセットで起動
```

```
// [b:GB62120] : 11 軸目の起動トリガ、ビットセットで起動
// [b:GB62130] : 12 軸目の起動トリガ、ビットセットで起動
// [b:GB62105] : 9 軸目のタッチパネル運転中フラグ
// [b:GB62115] : 10 軸目のタッチパネル運転中フラグ
// [b:GB62125] : 11 軸目のタッチパネル運転中フラグ
// [b:GB62135] : 12 軸目のタッチパネル運転中フラグ

// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 9 軸目
if([b:GB62100] == 1){
  [w:GD61000] = 9;

  // アラームチェック
  if([1-9:s32:400129] == 0){     // アラームが発生していない場合
    [b:GB62101] = 1;         // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;         // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                   // アラームが発生している場合
    [b:GB62101] = 0;         // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-9:s32:400201] == 0){     // 運転していない場合
    [b:GB62102] = 1;         // 運転中確認フラグ(1:運転していない)
  } else {                   // 運転中の場合
    [b:GB62102] = 0;         // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62101] == 1 && [b:GB62102] == 1){
    if([b:GB62105] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62100] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62103] = 1;     // 運転準備開始トリガ
      [b:GB62101] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62102] = 0;     // 運転中確認フラグのクリア
    } else {               // タッチパネル運転中なら、運転終了準備をする
      [b:GB62100] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62106] = 1;     // 運転終了開始トリガ
      [b:GB62101] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62102] = 0;     // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62100] = 0;     // このスクリプトの起動トリガを終了する
      [b:GB62101] = 0;     // アラーム発生状態確認フラグのクリア
      [b:GB62102] = 0;     // 運転中確認フラグのクリア
  }
}

// 10 軸目
if([b:GB62110] == 1){
  [w:GD61000] = 10;
```

```
  // アラームチェック
  if([1-10:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62111] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62111] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-10:s32:400201] == 0){   // 運転していない場合
    [b:GB62112] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62112] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62111] == 1 && [b:GB62112] == 1){
    if([b:GB62115] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62110] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62113] = 1;    // 運転準備開始トリガ
      [b:GB62111] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62112] = 0;    // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
      [b:GB62110] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62116] = 1;    // 運転終了開始トリガ
      [b:GB62111] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62112] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62110] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62111] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62112] = 0;    // 運転中確認フラグのクリア
  }
}

// 11 軸目
if([b:GB62120] == 1){
  [w:GD61000] = 11;

  // アラームチェック
  if([1-11:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62121] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62121] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-11:s32:400201] == 0){   // 運転していない場合
    [b:GB62122] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62122] = 0;        // 運転中確認フラグ(0:運転中)
```

```
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62121] == 1 && [b:GB62122] == 1){
    if([b:GB62125] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62120] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62123] = 1;    // 運転準備開始トリガ
      [b:GB62121] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62122] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62120] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62126] = 1;    // 運転終了開始トリガ
      [b:GB62121] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62122] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62120] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62121] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62122] = 0;    // 運転中確認フラグのクリア
  }
}

// 12 軸目
if([b:GB62130] == 1){
  [w:GD61000] = 12;

  // アラームチェック
  if([1-12:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62131] = 1;       // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;       // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                 // アラームが発生している場合
    [b:GB62131] = 0;       // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-12:s32:400201] == 0){   // 運転していない場合
    [b:GB62132] = 1;       // 運転中確認フラグ(1:運転していない)
  } else {                 // 運転中の場合
    [b:GB62132] = 0;       // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62131] == 1 && [b:GB62132] == 1){
    if([b:GB62135] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62130] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62133] = 1;    // 運転準備開始トリガ
      [b:GB62131] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62132] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62130] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62136] = 1;    // 運転終了開始トリガ
      [b:GB62131] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62132] = 0;    // 運転中確認フラグのクリア
```

```
        }
    } else {
        [b:GB62130] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62131] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62132] = 0;    // 運転中確認フラグのクリア
    }
}
```

スクリプト No.	31025	スクリプト名	Script31025
コメント	9-12_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62103

```
// タッチパネルから運転できるように準備するスクリプト
// NET-IN などの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 9-12 軸マルチ運転の画面スクリプト

// 9-12_マルチ運転準備スクリプト No.31025
// 9-12_マルチ運転開始_終了前確認スクリプト(No.31024)の GB62103、GB62113、GB62123、GB62133 の ON
中で、スクリプトを起動

// [b:GB62103],    [b:GB62113],    [b:GB62123],    [b:GB62133]   : 運転準備開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62104],    [b:GB62114],    [b:GB62124],    [b:GB62134]   : 運転準備中フラグ
// [b:GB62105],    [b:GB62115],    [b:GB62125],    [b:GB62135]   : タッチパネル運転中フラグ
// [u32:GD62002], [u32:GD62102], [u32:GD62202], [u32:GD62302] : NET-IN0～15 のバックアップデバイ
ス
// [u32:GD62034], [u32:GD62134], [u32:GD62234], [u32:GD62334] : IN0～7 のバックアップデバイス
// [s32:GD62050], [s32:GD62150], [s32:GD62250], [s32:GD62350] : モニタの指令位置のバックアップ
デバイス
// [s32:GD62052], [s32:GD62152], [s32:GD62252], [u32:GD62352] : プリセット位置のバックアップデ
バイス
// [u32:GD62054], [u32:GD62154], [u32:GD62254], [u32:GD62354] : JOG 移動量のバックアップデバイス
(GOT の最小移動量に使用)
// [w:GD62056],   [w:GD62156],    [w:GD62256],   [w:GD62356]     : Configuration 実行待ち時間用
タイマー

// 9 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62103] == 1 && [b:GB62104] == 0){

    [b:GB62104] = 1;                    // 運転準備中フラグ(1:運転準備中)
    bmov([1-9:u32:404449],[u32:GD62002],16);  // NET-IN0～15 のバックアップ
    bmov([1-9:u32:404353],[u32:GD62034],8); // IN0～7 のバックアップ
    [s32:GD62052] = [1-9:s32:400909];        // プリセット位置のバックアップ
    [u32:GD62054] = [1-9:u32:404169];        // JOG 移動量のバックアップ(GOT の最小移動量に使用)

    // NET-IN の入力値をクリアする
    [1-9:w:400126] = 0;

    // タッチパネルから運転する為に信号の割当を変更する
    [1-9:u32:404449] = 48; // NET-IN0  M0
    [1-9:u32:404451] = 49; // NET-IN1  M1
    [1-9:u32:404453] = 50; // NET-IN2  M2
    [1-9:u32:404455] = 51; // NET-IN3  M3
    [1-9:u32:404457] = 52; // NET-IN4  M4
    [1-9:u32:404459] = 53; // NET-IN5  M5
    [1-9:u32:404461] = 4;  // NET-IN6  START
    [1-9:u32:404463] = 3;  // NET-IN7  HOME
```

```
    [1-9:u32:404465] = 18;// NET-IN8  STOP
    [1-9:u32:404467] = 9; // NET-IN9  MS1
    [1-9:u32:404469] = 10;// NET-IN10 MS2
    [1-9:u32:404471] = 5; // NET-IN11 SSTART
    [1-9:u32:404473] = 6; // NET-IN12 +JOG
    [1-9:u32:404475] = 7; // NET-IN13 -JOG
    [1-9:u32:404477] = 1; // NET-IN14 FWD
    [1-9:u32:404479] = 2; // NET-IN15 RVS

    [1-9:u32:404353] = 32;   // IN0 R0
    [1-9:u32:404355] = 33;   // IN1 R1
    [1-9:u32:404357] = 34;   // IN2 R2
    [1-9:u32:404359] = 35;   // IN3 R3
    [1-9:u32:404361] = 36;   // IN4 R4
    [1-9:u32:404363] = 37;   // IN5 R5
    [1-9:u32:404365] = 18;   // IN6 STOP
    [1-9:u32:404367] = 39;   // IN7 R7

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD62050] = [1-9:s32:400199];
    [1-9:s32:400909] = [s32:GD62050];

    // Configuration の実行
    [1-9:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD62056] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62056] > 1 && [b:GB62104] == 1){

    [1-9:s32:400395] = 1;         // P-PRESET(プリセット)実行する
    [1-9:s32:400909] = [s32:GD62052];  // バックアップしたプリセット位置パラメータを元に戻す
    [1-9:s32:400397] = 0;         // Configuration 実行のゼロクリア
    [1-9:s32:400395] = 0;         // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62103] = 0;            // 運転準備開始トリガのクリア
                            // (このスクリプトの起動トリガを終了する)
    [b:GB62104] = 0;            // 運転準備中フラグのクリア
    [b:GB62105] = 1;            // タッチパネル運転中フラグ(1:運転中)
}

// 10 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62113] == 1 && [b:GB62114] == 0){

    [b:GB62114] = 1;                      // 運転準備中フラグ(1:運転準備中)
    bmov([1-10:u32:404449],[u32:GD62102],16);    // NET-IN0～15 のバックアップ
    bmov([1-10:u32:404353],[u32:GD62134],8);     // IN0～7 のバックアップ
    [s32:GD62152] = [1-10:s32:400909];      // プリセット位置のバックアップ
    [u32:GD62154] = [1-10:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

    // NET-IN の入力値をクリアする
    [1-10:w:400126] = 0;
```

```
  // タッチパネルから運転する為に信号の割当を変更する
  [1-10:u32:404449] = 48;     // NET-IN0  M0
  [1-10:u32:404451] = 49;     // NET-IN1  M1
  [1-10:u32:404453] = 50;     // NET-IN2  M2
  [1-10:u32:404455] = 51;     // NET-IN3  M3
  [1-10:u32:404457] = 52;     // NET-IN4  M4
  [1-10:u32:404459] = 53;     // NET-IN5  M5
  [1-10:u32:404461] = 4;   // NET-IN6  START
  [1-10:u32:404463] = 3;   // NET-IN7  HOME
  [1-10:u32:404465] = 18;     // NET-IN8  STOP
  [1-10:u32:404467] = 9;   // NET-IN9  MS1
  [1-10:u32:404469] = 10;     // NET-IN10 MS2
  [1-10:u32:404471] = 5;   // NET-IN11 SSTART
  [1-10:u32:404473] = 6;   // NET-IN12 +JOG
  [1-10:u32:404475] = 7;   // NET-IN13 -JOG
  [1-10:u32:404477] = 1;   // NET-IN14 FWD
  [1-10:u32:404479] = 2;   // NET-IN15 RVS

  [1-10:u32:404353] = 32;  // IN0 R0
  [1-10:u32:404355] = 33;  // IN1 R1
  [1-10:u32:404357] = 34;  // IN2 R2
  [1-10:u32:404359] = 35;  // IN3 R3
  [1-10:u32:404361] = 36;  // IN4 R4
  [1-10:u32:404363] = 37;  // IN5 R5
  [1-10:u32:404365] = 18;  // IN6 STOP
  [1-10:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62150] = [1-10:s32:400199];
  [1-10:s32:400909] = [s32:GD62150];

  // Configuration の実行
  [1-10:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD62156] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62156] > 1 && [b:GB62114] == 1){

  [1-10:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-10:s32:400909] = [s32:GD62152]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-10:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-10:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62113] = 0;              // 運転準備開始トリガのクリア
                              // (このスクリプトの起動トリガを終了する)
  [b:GB62114] = 0;              // 運転準備中フラグのクリア
  [b:GB62115] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 11 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62123] == 1 && [b:GB62124] == 0){
```

```
  [b:GB62124] = 1;                    // 運転準備中フラグ(1：運転準備中)
  bmov([1-11:u32:404449],[u32:GD62202],16);    // NET-IN0～15 のバックアップ
  bmov([1-11:u32:404353],[u32:GD62234],8);     // IN0～7 のバックアップ
  [s32:GD62252] = [1-11:s32:400909];      // プリセット位置のバックアップ
  [u32:GD62254] = [1-11:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-11:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-11:u32:404449] = 48;     // NET-IN0  M0
  [1-11:u32:404451] = 49;     // NET-IN1  M1
  [1-11:u32:404453] = 50;     // NET-IN2  M2
  [1-11:u32:404455] = 51;     // NET-IN3  M3
  [1-11:u32:404457] = 52;     // NET-IN4  M4
  [1-11:u32:404459] = 53;     // NET-IN5  M5
  [1-11:u32:404461] = 4;   // NET-IN6  START
  [1-11:u32:404463] = 3;   // NET-IN7  HOME
  [1-11:u32:404465] = 18;     // NET-IN8  STOP
  [1-11:u32:404467] = 9;   // NET-IN9  MS1
  [1-11:u32:404469] = 10;     // NET-IN10 MS2
  [1-11:u32:404471] = 5;   // NET-IN11 SSTART
  [1-11:u32:404473] = 6;   // NET-IN12 +JOG
  [1-11:u32:404475] = 7;   // NET-IN13 -JOG
  [1-11:u32:404477] = 1;   // NET-IN14 FWD
  [1-11:u32:404479] = 2;   // NET-IN15 RVS

  [1-11:u32:404353] = 32;  // IN0 R0
  [1-11:u32:404355] = 33;  // IN1 R1
  [1-11:u32:404357] = 34;  // IN2 R2
  [1-11:u32:404359] = 35;  // IN3 R3
  [1-11:u32:404361] = 36;  // IN4 R4
  [1-11:u32:404363] = 37;  // IN5 R5
  [1-11:u32:404365] = 18;  // IN6 STOP
  [1-11:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62250] = [1-11:s32:400199];
  [1-11:s32:400909] = [s32:GD62250];

  // Configuration の実行
  [1-11:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD62256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62256] > 1 && [b:GB62124] == 1){

  [1-11:s32:400395] = 1;           // P-PRESET(プリセット)実行する
  [1-11:s32:400909] = [s32:GD62252]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-11:s32:400397] = 0;           // Configuration 実行のゼロクリア
  [1-11:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62123] = 0;              // 運転準備開始トリガのクリア
```

```
                          //（このスクリプトの起動トリガを終了する）
  [b:GB62124] = 0;              // 運転準備中フラグのクリア
  [b:GB62125] = 1;              // タッチパネル運転中フラグ(1：運転中)
}

// 12 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62133] == 1 && [b:GB62134] == 0){

  [b:GB62134] = 1;                      // 運転準備中フラグ(1：運転準備中)
  bmov([1-12:u32:404449],[u32:GD62302],16);    // NET-IN0～15 のバックアップ
  bmov([1-12:u32:404353],[u32:GD62334],8);     // IN0～7 のバックアップ
  [s32:GD62352] = [1-12:s32:400909];       // プリセット位置のバックアップ
  [u32:GD62354] = [1-12:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-12:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-12:u32:404449] = 48;     // NET-IN0  M0
  [1-12:u32:404451] = 49;     // NET-IN1  M1
  [1-12:u32:404453] = 50;     // NET-IN2  M2
  [1-12:u32:404455] = 51;     // NET-IN3  M3
  [1-12:u32:404457] = 52;     // NET-IN4  M4
  [1-12:u32:404459] = 53;     // NET-IN5  M5
  [1-12:u32:404461] = 4;   // NET-IN6  START
  [1-12:u32:404463] = 3;   // NET-IN7  HOME
  [1-12:u32:404465] = 18;     // NET-IN8  STOP
  [1-12:u32:404467] = 9;   // NET-IN9  MS1
  [1-12:u32:404469] = 10;     // NET-IN10 MS2
  [1-12:u32:404471] = 5;   // NET-IN11 SSTART
  [1-12:u32:404473] = 6;   // NET-IN12 +JOG
  [1-12:u32:404475] = 7;   // NET-IN13 -JOG
  [1-12:u32:404477] = 1;   // NET-IN14 FWD
  [1-12:u32:404479] = 2;   // NET-IN15 RVS

  [1-12:u32:404353] = 32;  // IN0 R0
  [1-12:u32:404355] = 33;  // IN1 R1
  [1-12:u32:404357] = 34;  // IN2 R2
  [1-12:u32:404359] = 35;  // IN3 R3
  [1-12:u32:404361] = 36;  // IN4 R4
  [1-12:u32:404363] = 37;  // IN5 R5
  [1-12:u32:404365] = 18;  // IN6 STOP
  [1-12:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62350] = [1-12:s32:400199];
  [1-12:s32:400909] = [s32:GD62350];

  // Configuration の実行
  [1-12:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD62356] = [w:GS7];
}
```

```
// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62356] > 1 && [b:GB62134] == 1){

   [1-12:s32:400395] = 1;          // P-PRESET(プリセット)実行する
   [1-12:s32:400909] = [s32:GD62352]; // バックアップしたプリセット位置パラメータを元に戻す
   [1-12:s32:400397] = 0;          // Configuration 実行のゼロクリア
   [1-12:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62133] = 0;              // 運転準備開始トリガのクリア
                              //(このスクリプトの起動トリガを終了する)
   [b:GB62134] = 0;              // 運転準備中フラグのクリア
   [b:GB62135] = 1;              // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプト No.	31026	スクリプト名	Script31026
コメント	9-12_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62105

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データ No.を M0～M5(割付直した NET-IN)に反映させる
// 反映した M0～M5 は、ドライバ入力指令(400126)にて、ON する

// 9-12_マルチ運転中スクリプト No.31026
// 9-12_マルチ運転準備スクリプト(No.31025)の GB62105、GB62115、GB62125、GB62135 の ON 中で、スクリプトを起動

// [b:GB62105], [b:GB62115], [b:GB62125], [b:GB62135] : タッチパネル運転中フラグ(このスクリプトの起動トリガ)
//また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-9:u32:400195] ～ [1-12:u32:400195] : モニタの現在の選択データ No.の Modbus アドレス
// [u32:GD62000], [u32GD62100], [GD62200], [u32:GD62300] : 運転データ No.のデバイス

// 選択した運転データ No.をドライバ入力指令の M0～M5 に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データ No.(400195)と選択した運転データ No.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

// 9 軸目
if([b:GB62105] == 1){

   if([1-9:u32:400195] != [u32:GD62000]){
      [1-9:b:400126.b0] = [b:GD62000.b0];     // M0
      [1-9:b:400126.b1] = [b:GD62000.b1];     // M1
      [1-9:b:400126.b2] = [b:GD62000.b2];     // M2
      [1-9:b:400126.b3] = [b:GD62000.b3];     // M3
      [1-9:b:400126.b4] = [b:GD62000.b4];     // M4
      [1-9:b:400126.b5] = [b:GD62000.b5];     // M5
   }
}

// 10 軸目
if([b:GB62115] == 1){

   if([1-10:u32:400195] != [u32:GD62100]){
      [1-10:b:400126.b0] = [b:GD62100.b0]; // M0
      [1-10:b:400126.b1] = [b:GD62100.b1]; // M1
      [1-10:b:400126.b2] = [b:GD62100.b2]; // M2
```

```
      [1-10:b:400126.b3] = [b:GD62100.b3]; // M3
      [1-10:b:400126.b4] = [b:GD62100.b4]; // M4
      [1-10:b:400126.b5] = [b:GD62100.b5]; // M5
   }
}

// 11 軸目
if([b:GB62125] == 1){

   if([1-11:u32:400195] != [u32:GD62200]){
      [1-11:b:400126.b0] = [b:GD62200.b0]; // M0
      [1-11:b:400126.b1] = [b:GD62200.b1]; // M1
      [1-11:b:400126.b2] = [b:GD62200.b2]; // M2
      [1-11:b:400126.b3] = [b:GD62200.b3]; // M3
      [1-11:b:400126.b4] = [b:GD62200.b4]; // M4
      [1-11:b:400126.b5] = [b:GD62200.b5]; // M5
   }
}

// 12 軸目
if([b:GB62135] == 1){

   if([1-12:u32:400195] != [u32:GD62300]){
      [1-12:b:400126.b0] = [b:GD62300.b0]; // M0
      [1-12:b:400126.b1] = [b:GD62300.b1]; // M1
      [1-12:b:400126.b2] = [b:GD62300.b2]; // M2
      [1-12:b:400126.b3] = [b:GD62300.b3]; // M3
      [1-12:b:400126.b4] = [b:GD62300.b4]; // M4
      [1-12:b:400126.b5] = [b:GD62300.b5]; // M5
   }
}
```

スクリプトNo.	31027	スクリプト名	Script31027
コメント	9-12_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62106

```
// タッチパネルからの運転を終了するスクリプト
// バックアップしたNET-INなどの信号を元に戻したり、必要なパラメータなどを元に戻す
// 9-12軸のマルチ運転の画面スクリプト

// 9-12_マルチ運転終了スクリプト No.31027
// 9-12_マルチ運転開始_終了前確認スクリプト(No.31024)の GB62106、GB62116、GB62126、GB62136 の ON
中で、スクリプトを起動

// [b:GB62105],    [b:GB62115],    [b:GB62125],    [b:GB62135]    : タッチパネル運転中フラグ
// [b:GB62106],    [b:GB62116],    [b:GB62126],    [b:GB62136]    : 運転終了開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62107],    [b:GB62117],    [b:GB62127],    [b:GB62137]    : 運転終了中フラグ
// [u32:GD62002], [u32:GD62102], [u32:GD62202], [u32:GD62302] : NET-IN0～15のバックアップデバイ
ス
// [u32:GD62034], [u32:GD62134], [u32:GD62234], [u32:GD62334] : IN0～7のバックアップデバイス
// [s32:GD62050], [s32:GD62150], [s32:GD62250], [s32:GD62350] : モニタの指令位置のバックアップ
デバイス
// [s32:GD62052], [s32:GD62152], [s32:GD62252], [s32:GD62352] : プリセット位置のバックアップデ
バイス
// [u32:GD62054], [u32:GD62154], [u32:GD62254], [u32:GD62354] : JOG移動量のバックアップデバイス
(GOTの最小移動量に使用)
// [w:GD62056],   [w:GD62156],    [w:GD62256],   [w:GD62356]      : Configuration 実行待ち時間用
```

```
タイマー

// 9軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62106] == 1 && [b:GB62107] == 0){

  [b:GB62105] = 0;                 // タッチパネル運転中フラグのクリア
  [b:GB62107] = 1;                 // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD62002],[1-9:u32:404449],16);  // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD62034],[1-9:u32:404353],8); // バックアップした IN0～7 を元に戻す
  [1-9:u32:404169] = [u32:GD62054];        // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62050] = [1-9:s32:400199];
  [1-9:s32:400909] = [s32:GD62050];

  // NET-IN の入力値をクリアする
  [1-9:w:400126] = 0;

  // Configuration の実行
  [1-9:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD62056] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62056] > 1 && [b:GB62107] == 1){

  [1-9:s32:400395] = 1;            // P-PRESET(プリセット)実行する
  [1-9:s32:400909] = [s32:GD62052];     // プリセット位置パラメータを元に戻す
  [1-9:s32:400397] = 0;            // Configuration 実行のゼロクリア
  [1-9:s32:400395] = 0;            // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62106] = 0;               // 運転終了開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
  [b:GB62107] = 0;               // 運転終了中フラグのクリア
}

// 10軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62116] == 1 && [b:GB62117] == 0){

  [b:GB62115] = 0;                     // タッチパネル運転中フラグのクリア
  [b:GB62117] = 1;                     // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD62102],[1-10:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD62134],[1-10:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-10:u32:404169] = [u32:GD62154];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62150] = [1-10:s32:400199];
  [1-10:s32:400909] = [s32:GD62150];

  // NET-IN の入力値をクリアする
  [1-10:w:400126] = 0;
```

```
  // Configuration の実行
  [1-10:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD62156] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62156] > 1 && [b:GB62117] == 1){

  [1-10:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-10:s32:400909] = [s32:GD62152]; // プリセット位置パラメータを元に戻す
  [1-10:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-10:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62116] = 0;              // 運転終了開始トリガのクリア
                             // (このスクリプトの起動トリガを終了する)
  [b:GB62117] = 0;              // 運転終了中フラグのクリア
}

// 11 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62126] == 1 && [b:GB62127] == 0){

  [b:GB62125] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62127] = 1;                    // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD62202],[1-11:u32:404449],16);   // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD62234],[1-11:u32:404353],8);    // バックアップした IN0～7 を元に戻す
  [1-11:u32:404169] = [u32:GD62254];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62250] = [1-11:s32:400199];
  [1-11:s32:400909] = [s32:GD62250];

  // NET-IN の入力値をクリアする
  [1-11:w:400126] = 0;

  // Configuration の実行
  [1-11:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)-
  [w:GD62256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62256] > 1 && [b:GB62127] == 1){

  [1-11:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-11:s32:400909] = [s32:GD62252]; // プリセット位置パラメータを元に戻す
  [1-11:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-11:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62126] = 0;              // 運転終了開始トリガのクリア
                             // (このスクリプトの起動トリガを終了する)
  [b:GB62127] = 0;              // 運転終了中フラグのクリア
}
```

```
// 12軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62136] == 1 && [b:GB62137] == 0){

  [b:GB62135] = 0;                      // タッチパネル運転中フラグのクリア
  [b:GB62137] = 1;                      // 運転終了中フラグ(1：運転終了中)
  bmov([u32:GD62302],[1-12:u32:404449],16);    // バックアップしたNET-IN0～15を元に戻す
  bmov([u32:GD62334],[1-12:u32:404353],8);     // バックアップしたIN0～7を元に戻す
  [1-12:u32:404169] = [u32:GD62354];      // バックアップしたJOG移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62350] = [1-12:s32:400199];
  [1-12:s32:400909] = [s32:GD62350];

  // NET-INの入力値をクリアする
  [1-12:w:400126] = 0;

  // Configurationの実行
  [1-12:s32:400397] = 1;

  // Configuration実行待ち時間のタイマー記録(単位は秒)
  [w:GD62356] = [w:GS7];
}

// Configuration実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62356] > 1 && [b:GB62137] == 1){

  [1-12:s32:400395] = 1;            // P-PRESET(プリセット)実行する
  [1-12:s32:400909] = [s32:GD62352]; // プリセット位置パラメータを元に戻す
  [1-12:s32:400397] = 0;            // Configuration実行のゼロクリア
  [1-12:s32:400395] = 0;            // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62136] = 0;               // 運転終了開始トリガのクリア
                                 // (このスクリプトの起動トリガを終了する)
  [b:GB62137] = 0;               // 運転終了中フラグのクリア
}
```

スクリプトNo.	31024	スクリプト名	Script31024
コメント	9-12_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62110

ベース画面31024のスクリプトNo.31024と同じです。

スクリプトNo.	31025	スクリプト名	Script31025
コメント	9-12_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62113

ベース画面31024のスクリプトNo.31025と同じです。

スクリプトNo.	31026	スクリプト名	Script31026
コメント	9-12_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62115

ベース画面31024のスクリプトNo.31026と同じです。

スクリプトNo.	31027	スクリプト名	Script31027
コメント	9-12_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62116

ベース画面 31024 のスクリプト No.31027 と同じです。			
スクリプト No.	31024	スクリプト名	Script31024
コメント	9-12_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62120
ベース画面 31024 のスクリプト No.31024 と同じです。			
スクリプト No.	31025	スクリプト名	Script31025
コメント	9-12_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62123
ベース画面 31024 のスクリプト No.31025 と同じです。			
スクリプト No.	31026	スクリプト名	Script31026
コメント	9-12_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62125
ベース画面 31024 のスクリプト No.31026 と同じです。			
スクリプト No.	31027	スクリプト名	Script31027
コメント	9-12_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62126
ベース画面 31024 のスクリプト No.31027 と同じです。			
スクリプト No.	31024	スクリプト名	Script31024
コメント	9-12_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62130
ベース画面 31024 のスクリプト No.31024 と同じです。			
スクリプト No.	31025	スクリプト名	Script31025
コメント	9-12_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62133
ベース画面 31024 のスクリプト No.31025 と同じです。			
スクリプト No.	31026	スクリプト名	Script31026
コメント	9-12_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62135
ベース画面 31024 のスクリプト No.31026 と同じです。			
スクリプト No.	31027	スクリプト名	Script31027
コメント	9-12_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62136
ベース画面 31024 のスクリプト No.31027 と同じです。			

ベース画面 31025

スクリプト No.	31028	スクリプト名	Script31028
コメント	13-16_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62140

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 13-16 軸のマルチ運転の画面スクリプト

// 13-16_マルチ運転開始_終了前確認スクリプト No.31028
// 13-16 軸マルチ運転画面のそれぞれの開始ボタンの ON 中周期、2 秒で、スクリプトを起動

// [b:GB62140] : 13 軸目の起動トリガ、ビットセットで起動
// [b:GB62150] : 14 軸目の起動トリガ、ビットセットで起動
// [b:GB62160] : 15 軸目の起動トリガ、ビットセットで起動
// [b:GB62170] : 16 軸目の起動トリガ、ビットセットで起動
// [b:GB62145] : 13 軸目のタッチパネル運転中フラグ
// [b:GB62155] : 14 軸目のタッチパネル運転中フラグ
// [b:GB62165] : 15 軸目のタッチパネル運転中フラグ
// [b:GB62175] : 16 軸目のタッチパネル運転中フラグ

// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 13 軸目
if([b:GB62140] == 1){
  [w:GD61000] = 13;

  // アラームチェック
  if([1-13:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62141] = 1;       // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;       // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                 // アラームが発生している場合
    [b:GB62141] = 0;       // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-13:s32:400201] == 0){   // 運転していない場合
    [b:GB62142] = 1;       // 運転中確認フラグ(1:運転していない)
  } else {                 // 運転中の場合
    [b:GB62142] = 0;       // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62141] == 1 && [b:GB62142] == 1){
    if([b:GB62145] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62140] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62143] = 1;    // 運転準備開始トリガ
      [b:GB62141] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62142] = 0;    // 運転中確認フラグのクリア
    } else {               // タッチパネル運転中なら、運転終了準備をする
      [b:GB62140] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62146] = 1;    // 運転終了開始トリガ
```

```
        [b:GB62141] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62142] = 0;    // 運転中確認フラグのクリア
      }
   } else {
        [b:GB62140] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62141] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62142] = 0;    // 運転中確認フラグのクリア
   }
}

// 14軸目
if([b:GB62150] == 1){
   [w:GD61000] = 14;

   // アラームチェック
   if([1-14:s32:400129] == 0){   // アラームが発生していない場合
      [b:GB62151] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
      [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
   } else {                   // アラームが発生している場合
      [b:GB62151] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
      [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
   }

   // 運転中チェック
   if([1-14:s32:400201] == 0){   // 運転していない場合
      [b:GB62152] = 1;        // 運転中確認フラグ(1:運転していない)
   } else {                   // 運転中の場合
      [b:GB62152] = 0;        // 運転中確認フラグ(0:運転中)
      [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
   }

   // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
   if([b:GB62151] == 1 && [b:GB62152] == 1){
      if([b:GB62155] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
        [b:GB62150] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62153] = 1;    // 運転準備開始トリガ
        [b:GB62151] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62152] = 0;    // 運転中確認フラグのクリア
      } else {                // タッチパネル運転中なら、運転終了準備をする
        [b:GB62150] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62156] = 1;    // 運転終了開始トリガ
        [b:GB62151] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62152] = 0;    // 運転中確認フラグのクリア
      }
   } else {
        [b:GB62150] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62151] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62152] = 0;    // 運転中確認フラグのクリア
   }
}

// 15軸目
if([b:GB62160] == 1){
   [w:GD61000] = 15;

   // アラームチェック
```

```
  if([1-15:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62161] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62161] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-15:s32:400201] == 0){   // 運転していない場合
    [b:GB62162] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62162] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62161] == 1 && [b:GB62162] == 1){
    if([b:GB62165] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62160] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62163] = 1;    // 運転準備開始トリガ
      [b:GB62161] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62162] = 0;    // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
      [b:GB62160] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62166] = 1;    // 運転終了開始トリガ
      [b:GB62161] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62162] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62160] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62161] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62162] = 0;    // 運転中確認フラグのクリア
  }
}

// 16軸目
if([b:GB62170] == 1){
  [w:GD61000] = 16;

  // アラームチェック
  if([1-16:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62171] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62171] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-16:s32:400201] == 0){   // 運転していない場合
    [b:GB62172] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62172] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }
```

```
  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62171] == 1 && [b:GB62172] == 1){
    if([b:GB62175] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62170] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62173] = 1;    // 運転準備開始トリガ
      [b:GB62171] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62172] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62170] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62176] = 1;    // 運転終了開始トリガ
      [b:GB62171] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62172] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62170] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62171] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62172] = 0;    // 運転中確認フラグのクリア
  }
}
```

スクリプトNo.	31029	スクリプト名	Script31029
コメント	13-16_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62143

```
// タッチパネルから運転できるように準備するスクリプト
// NET-INなどの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 13-16軸マルチ運転の画面スクリプト

// 13-16_マルチ運転準備スクリプト No.31029
// 13-16_マルチ運転開始_終了前確認スクリプト(No.31028)の GB62143、GB62153、GB62163、GB62173 の ON
中で、スクリプトを起動

// [b:GB62143],    [b:GB62153],    [b:GB62163],    [b:GB62173]   : 運転準備開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62144],    [b:GB62154],    [b:GB62164],    [b:GB62174]   : 運転準備中フラグ
// [b:GB62145],    [b:GB62155],    [b:GB62165],    [b:GB62175]   : タッチパネル運転中フラグ
// [u32:GD62402], [u32:GD62502], [u32:GD62602], [u32:GD62702] : NET-IN0～15のバックアップデバイ
ス
// [u32:GD62434], [u32:GD62534], [u32:GD62634], [u32:GD62734] : IN0～7のバックアップデバイス
// [s32:GD62450], [s32:GD62550], [s32:GD62650], [s32:GD62750] : モニタの指令位置のバックアップ
デバイス
// [s32:GD62452], [s32:GD62552], [s32:GD62652], [u32:GD62752] : プリセット位置のバックアップデ
バイス
// [u32:GD62454], [u32:GD62554], [u32:GD62654], [u32:GD62754] : JOG移動量のバックアップデバイス
(GOTの最小移動量に使用)
// [w:GD62456],   [w:GD62556],    [w:GD62656],   [w:GD62756]     : Configuration 実行待ち時間用
タイマー

// 13軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62143] == 1 && [b:GB62144] == 0){

  [b:GB62144] = 1;                       // 運転準備中フラグ(1:運転準備中)
  bmov([1-13:u32:404449],[u32:GD62402],16);     // NET-IN0～15のバックアップ
  bmov([1-13:u32:404353],[u32:GD62434],8);      // IN0～7のバックアップ
  [s32:GD62452] = [1-13:s32:400909];        // プリセット位置のバックアップ
  [u32:GD62454] = [1-13:u32:404169];        // JOG移動量のバックアップ(GOTの最小移動量に使用)
```

```
    // NET-INの入力値をクリアする
    [1-13:w:400126] = 0;

    // タッチパネルから運転する為に信号の割当を変更する
    [1-13:u32:404449] = 48;     // NET-IN0  M0
    [1-13:u32:404451] = 49;     // NET-IN1  M1
    [1-13:u32:404453] = 50;     // NET-IN2  M2
    [1-13:u32:404455] = 51;     // NET-IN3  M3
    [1-13:u32:404457] = 52;     // NET-IN4  M4
    [1-13:u32:404459] = 53;     // NET-IN5  M5
    [1-13:u32:404461] = 4;   // NET-IN6  START
    [1-13:u32:404463] = 3;   // NET-IN7  HOME
    [1-13:u32:404465] = 18;     // NET-IN8  STOP
    [1-13:u32:404467] = 9;   // NET-IN9  MS1
    [1-13:u32:404469] = 10;     // NET-IN10 MS2
    [1-13:u32:404471] = 5;   // NET-IN11 SSTART
    [1-13:u32:404473] = 6;   // NET-IN12 +JOG
    [1-13:u32:404475] = 7;   // NET-IN13 -JOG
    [1-13:u32:404477] = 1;   // NET-IN14 FWD
    [1-13:u32:404479] = 2;   // NET-IN15 RVS

    [1-13:u32:404353] = 32;  // IN0 R0
    [1-13:u32:404355] = 33;  // IN1 R1
    [1-13:u32:404357] = 34;  // IN2 R2
    [1-13:u32:404359] = 35;  // IN3 R3
    [1-13:u32:404361] = 36;  // IN4 R4
    [1-13:u32:404363] = 37;  // IN5 R5
    [1-13:u32:404365] = 18;  // IN6 STOP
    [1-13:u32:404367] = 39;  // IN7 R7

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD62450] = [1-13:s32:400199];
    [1-13:s32:400909] = [s32:GD62450];

    // Configurationの実行
    [1-13:s32:400397] = 1;

    // Configuration実行待ち時間のタイマー記録(単位は秒)
    [w:GD62456] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62456] > 1 && [b:GB62144] == 1){

    [1-13:s32:400395] = 1;           // P-PRESET(プリセット)実行する
    [1-13:s32:400909] = [s32:GD62452]; // バックアップしたプリセット位置パラメータを元に戻す
    [1-13:s32:400397] = 0;           // Configuration実行のゼロクリア
    [1-13:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62143] = 0;              // 運転準備開始トリガのクリア
                             // (このスクリプトの起動トリガを終了する)
    [b:GB62144] = 0;              // 運転準備中フラグのクリア
    [b:GB62145] = 1;              // タッチパネル運転中フラグ(1:運転中)
}
```

```
// 14軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62153] == 1 && [b:GB62154] == 0){

   [b:GB62154] = 1;                     // 運転準備中フラグ(1:運転準備中)
   bmov([1-14:u32:404449],[u32:GD62502],16);    // NET-IN0～15のバックアップ
   bmov([1-14:u32:404353],[u32:GD62534],8);     // IN0～7のバックアップ
   [s32:GD62552] = [1-14:s32:400909];      // プリセット位置のバックアップ
   [u32:GD62554] = [1-14:u32:404169];      // JOG移動量のバックアップ(GOTの最小移動量に使用)

   // NET-INの入力値をクリアする
   [1-14:w:400126] = 0;

   // タッチパネルから運転する為に信号の割当を変更する
   [1-14:u32:404449] = 48;     // NET-IN0  M0
   [1-14:u32:404451] = 49;     // NET-IN1  M1
   [1-14:u32:404453] = 50;     // NET-IN2  M2
   [1-14:u32:404455] = 51;     // NET-IN3  M3
   [1-14:u32:404457] = 52;     // NET-IN4  M4
   [1-14:u32:404459] = 53;     // NET-IN5  M5
   [1-14:u32:404461] = 4;   // NET-IN6  START
   [1-14:u32:404463] = 3;   // NET-IN7  HOME
   [1-14:u32:404465] = 18;     // NET-IN8  STOP
   [1-14:u32:404467] = 9;   // NET-IN9  MS1
   [1-14:u32:404469] = 10;     // NET-IN10 MS2
   [1-14:u32:404471] = 5;   // NET-IN11 SSTART
   [1-14:u32:404473] = 6;   // NET-IN12 +JOG
   [1-14:u32:404475] = 7;   // NET-IN13 -JOG
   [1-14:u32:404477] = 1;   // NET-IN14 FWD
   [1-14:u32:404479] = 2;   // NET-IN15 RVS

   [1-14:u32:404353] = 32;  // IN0 R0
   [1-14:u32:404355] = 33;  // IN1 R1
   [1-14:u32:404357] = 34;  // IN2 R2
   [1-14:u32:404359] = 35;  // IN3 R3
   [1-14:u32:404361] = 36;  // IN4 R4
   [1-14:u32:404363] = 37;  // IN5 R5
   [1-14:u32:404365] = 18;  // IN6 STOP
   [1-14:u32:404367] = 39;  // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD62550] = [1-14:s32:400199];
   [1-14:s32:400909] = [s32:GD62550];

   // Configurationの実行
   [1-14:s32:400397] = 1;

   // Configuration実行待ち時間のタイマー記録(単位は秒)
   [w:GD62556] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62556] > 1 && [b:GB62154] == 1){

   [1-14:s32:400395] = 1;          // P-PRESET(プリセット)実行する
   [1-14:s32:400909] = [s32:GD62552]; // バックアップしたプリセット位置パラメータを元に戻す
```

```
  [1-14:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-14:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62153] = 0;              // 運転準備開始トリガのクリア
                              // (このスクリプトの起動トリガを終了する)
  [b:GB62154] = 0;              // 運転準備中フラグのクリア
  [b:GB62155] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 15 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62163] == 1 && [b:GB62164] == 0){

  [b:GB62164] = 1;                    // 運転準備中フラグ(1:運転準備中)
  bmov([1-15:u32:404449],[u32:GD62602],16);    // NET-IN0～15 のバックアップ
  bmov([1-15:u32:404353],[u32:GD62634],8);     // IN0～7 のバックアップ
  [s32:GD62652] = [1-15:s32:400909];      // プリセット位置のバックアップ
  [u32:GD62654] = [1-15:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-15:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-15:u32:404449] = 48;     // NET-IN0  M0
  [1-15:u32:404451] = 49;     // NET-IN1  M1
  [1-15:u32:404453] = 50;     // NET-IN2  M2
  [1-15:u32:404455] = 51;     // NET-IN3  M3
  [1-15:u32:404457] = 52;     // NET-IN4  M4
  [1-15:u32:404459] = 53;     // NET-IN5  M5
  [1-15:u32:404461] = 4;   // NET-IN6  START
  [1-15:u32:404463] = 3;   // NET-IN7  HOME
  [1-15:u32:404465] = 18;     // NET-IN8  STOP
  [1-15:u32:404467] = 9;   // NET-IN9  MS1
  [1-15:u32:404469] = 10;     // NET-IN10 MS2
  [1-15:u32:404471] = 5;   // NET-IN11 SSTART
  [1-15:u32:404473] = 6;   // NET-IN12 +JOG
  [1-15:u32:404475] = 7;   // NET-IN13 -JOG
  [1-15:u32:404477] = 1;   // NET-IN14 FWD
  [1-15:u32:404479] = 2;   // NET-IN15 RVS

  [1-15:u32:404353] = 32;  // IN0 R0
  [1-15:u32:404355] = 33;  // IN1 R1
  [1-15:u32:404357] = 34;  // IN2 R2
  [1-15:u32:404359] = 35;  // IN3 R3
  [1-15:u32:404361] = 36;  // IN4 R4
  [1-15:u32:404363] = 37;  // IN5 R5
  [1-15:u32:404365] = 18;  // IN6 STOP
  [1-15:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62650] = [1-15:s32:400199];
  [1-15:s32:400909] = [s32:GD62650];

  // Configuration の実行
  [1-15:s32:400397] = 1;
```

```
    // Configuration実行待ち時間のタイマー記録(単位は秒)
    [w:GD62656] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62656] > 1 && [b:GB62164] == 1){

    [1-15:s32:400395] = 1;          // P-PRESET(プリセット)実行する
    [1-15:s32:400909] = [s32:GD62652]; // バックアップしたプリセット位置パラメータを元に戻す
    [1-15:s32:400397] = 0;          // Configuration実行のゼロクリア
    [1-15:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62163] = 0;              // 運転準備開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
    [b:GB62164] = 0;              // 運転準備中フラグのクリア
    [b:GB62165] = 1;              // タッチパネル運転中フラグ(1：運転中)
}

// 16軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62173] == 1 && [b:GB62174] == 0){

    [b:GB62174] = 1;                    // 運転準備中フラグ(1：運転準備中)
    bmov([1-16:u32:404449],[u32:GD62702],16);    // NET-IN0～15のバックアップ
    bmov([1-16:u32:404353],[u32:GD62734],8);     // IN0～7のバックアップ
    [s32:GD62752] = [1-16:s32:400909];       // プリセット位置のバックアップ
    [u32:GD62754] = [1-16:u32:404169];       // JOG移動量のバックアップ(GOTの最小移動量に使用)

    // NET-INの入力値をクリアする
    [1-16:w:400126] = 0;

    // タッチパネルから運転する為に信号の割当を変更する
    [1-16:u32:404449] = 48;     // NET-IN0  M0
    [1-16:u32:404451] = 49;     // NET-IN1  M1
    [1-16:u32:404453] = 50;     // NET-IN2  M2
    [1-16:u32:404455] = 51;     // NET-IN3  M3
    [1-16:u32:404457] = 52;     // NET-IN4  M4
    [1-16:u32:404459] = 53;     // NET-IN5  M5
    [1-16:u32:404461] = 4;   // NET-IN6  START
    [1-16:u32:404463] = 3;   // NET-IN7  HOME
    [1-16:u32:404465] = 18;     // NET-IN8  STOP
    [1-16:u32:404467] = 9;   // NET-IN9  MS1
    [1-16:u32:404469] = 10;     // NET-IN10 MS2
    [1-16:u32:404471] = 5;   // NET-IN11 SSTART
    [1-16:u32:404473] = 6;   // NET-IN12 +JOG
    [1-16:u32:404475] = 7;   // NET-IN13 -JOG
    [1-16:u32:404477] = 1;   // NET-IN14 FWD
    [1-16:u32:404479] = 2;   // NET-IN15 RVS

    [1-16:u32:404353] = 32;  // IN0 R0
    [1-16:u32:404355] = 33;  // IN1 R1
    [1-16:u32:404357] = 34;  // IN2 R2
    [1-16:u32:404359] = 35;  // IN3 R3
    [1-16:u32:404361] = 36;  // IN4 R4
    [1-16:u32:404363] = 37;  // IN5 R5
    [1-16:u32:404365] = 18;  // IN6 STOP
```

```
   [1-16:u32:404367] = 39;  // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD62750] = [1-16:s32:400199];
   [1-16:s32:400909] = [s32:GD62750];

   // Configurationの実行
   [1-16:s32:400397] = 1;

   // Configuration実行待ち時間のタイマー記録(単位は秒)
   [w:GD62756] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62756] > 1 && [b:GB62174] == 1){

   [1-16:s32:400395] = 1;           // P-PRESET(プリセット)実行する
   [1-16:s32:400909] = [s32:GD62752]; // バックアップしたプリセット位置パラメータを元に戻す
   [1-16:s32:400397] = 0;           // Configuration実行のゼロクリア
   [1-16:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62173] = 0;              // 運転準備開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
   [b:GB62174] = 0;              // 運転準備中フラグのクリア
   [b:GB62175] = 1;              // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプトNo.	31030	スクリプト名	Script31030
コメント	13-16_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62145

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データNo.をM0～M5(割付直したNET-IN)に反映させる
// 反映したM0～M5は、ドライバ入力指令(400126)にて、ONする

// 13-16_マルチ運転中スクリプト No.31030
//13-16_マルチ運転準備スクリプト(No.31029)のGB62145、GB62155、GB62165、GB62175のON中で、スクリプトを起動

// [b:GB62145], [b:GB62155], [b:GB62165], [b:GB62175] : タッチパネル運転中フラグ(このスクリプトの起動トリガ)
//また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-13:u32:400195] ～ [1-16:u32:400195] : モニタの現在の選択データNo.のModbusアドレス
// [u32:GD62400], [u32GD62500], [GD62600], [u32:GD62700] : 運転データNo.のデバイス

// 選択した運転データNo.をドライバ入力指令のM0～M5に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データNo.(400195)と選択した運転データNo.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

// 13軸目
if([b:GB62145] == 1){

   if([1-13:u32:400195] != [u32:GD62400]){
      [1-13:b:400126.b0] = [b:GD62400.b0];    // M0
      [1-13:b:400126.b1] = [b:GD62400.b1];    // M1
      [1-13:b:400126.b2] = [b:GD62400.b2];    // M2
```

```
      [1-13:b:400126.b3] = [b:GD62400.b3];    // M3
      [1-13:b:400126.b4] = [b:GD62400.b4];    // M4
      [1-13:b:400126.b5] = [b:GD62400.b5];    // M5
   }
}

// 14軸目
if([b:GB62155] == 1){

   if([1-14:u32:400195] != [u32:GD62500]){
      [1-14:b:400126.b0] = [b:GD62500.b0];    // M0
      [1-14:b:400126.b1] = [b:GD62500.b1];    // M1
      [1-14:b:400126.b2] = [b:GD62500.b2];    // M2
      [1-14:b:400126.b3] = [b:GD62500.b3];    // M3
      [1-14:b:400126.b4] = [b:GD62500.b4];    // M4
      [1-14:b:400126.b5] = [b:GD62500.b5];    // M5
   }
}

// 15軸目
if([b:GB62165] == 1){

   if([1-15:u32:400195] != [u32:GD62600]){
      [1-15:b:400126.b0] = [b:GD62600.b0];    // M0
      [1-15:b:400126.b1] = [b:GD62600.b1];    // M1
      [1-15:b:400126.b2] = [b:GD62600.b2];    // M2
      [1-15:b:400126.b3] = [b:GD62600.b3];    // M3
      [1-15:b:400126.b4] = [b:GD62600.b4];    // M4
      [1-15:b:400126.b5] = [b:GD62600.b5];    // M5
   }
}

// 16軸目
if([b:GB62175] == 1){

   if([1-16:u32:400195] != [u32:GD62700]){
      [1-16:b:400126.b0] = [b:GD62700.b0];    // M0
      [1-16:b:400126.b1] = [b:GD62700.b1];    // M1
      [1-16:b:400126.b2] = [b:GD62700.b2];    // M2
      [1-16:b:400126.b3] = [b:GD62700.b3];    // M3
      [1-16:b:400126.b4] = [b:GD62700.b4];    // M4
      [1-16:b:400126.b5] = [b:GD62700.b5];    // M5
   }
}
```

スクリプトNo.	31031	スクリプト名	Script31031
コメント	13-16_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62146

```
// タッチパネルからの運転を終了するスクリプト
// バックアップしたNET-INなどの信号を元に戻したり、必要なパラメータなどを元に戻す
// 13-16軸のマルチ運転の画面スクリプト

// 13-16_マルチ運転終了スクリプト No.31031
// 13-16_マルチ運転開始_終了前確認スクリプト(No.31028)の GB62146、GB62156、GB62166、GB62176 の ON
中で、スクリプトを起動

// [b:GB62145],    [b:GB62155],    [b:GB62165],    [b:GB62175]   :  タッチパネル運転中フラグ
```

```
// [b:GB62146],    [b:GB62156],    [b:GB62166],    [b:GB62176]   : 運転終了開始トリガ(このスクリプトの起動トリガ)
// [b:GB62147],    [b:GB62157],    [b:GB62167],    [b:GB62177]   : 運転終了中フラグ
// [u32:GD62402], [u32:GD62502], [u32:GD62602], [u32:GD62702] : NET-IN0～15 のバックアップデバイス
// [u32:GD62434], [u32:GD62534], [u32:GD62634], [u32:GD62734] : IN0～7 のバックアップデバイス
// [s32:GD62450], [s32:GD62550], [s32:GD62650], [s32:GD62750] : モニタの指令位置のバックアップデバイス
// [s32:GD62452], [s32:GD62552], [s32:GD62652], [s32:GD62752] : プリセット位置のバックアップデバイス
// [u32:GD62454], [u32:GD62554], [u32:GD62654], [u32:GD62754] : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD62456],   [w:GD62556],   [w:GD62656],   [w:GD62756]    : Configuration 実行待ち時間用タイマー

// 13 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62146] == 1 && [b:GB62147] == 0){

   [b:GB62145] = 0;                    // タッチパネル運転中フラグのクリア
   [b:GB62147] = 1;                    // 運転終了中フラグ(1：運転終了中)
   bmov([u32:GD62402],[1-13:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
   bmov([u32:GD62434],[1-13:u32:404353],8);     // バックアップした IN0～7 を元に戻す
   [1-13:u32:404169] = [u32:GD62454];      // バックアップした JOG 移動量を元に戻す

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD62450] = [1-13:s32:400199];
   [1-13:s32:400909] = [s32:GD62450];

   // NET-IN の入力値をクリアする
   [1-13:w:400126] = 0;

   // Configuration の実行
   [1-13:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD62456] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62456] > 1 && [b:GB62147] == 1){

   [1-13:s32:400395] = 1;           // P-PRESET(プリセット)実行する
   [1-13:s32:400909] = [s32:GD62452]; // プリセット位置パラメータを元に戻す
   [1-13:s32:400397] = 0;           // Configuration 実行のゼロクリア
   [1-13:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62146] = 0;              // 運転終了開始トリガのクリア
                            // (このスクリプトの起動トリガを終了する)
   [b:GB62147] = 0;              // 運転終了中フラグのクリア
}

// 14 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62156] == 1 && [b:GB62157] == 0){
```

```
    [b:GB62155] = 0;                    // タッチパネル運転中フラグのクリア
    [b:GB62157] = 1;                    // 運転終了中フラグ(1：運転終了中)
    bmov([u32:GD62502],[1-14:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
    bmov([u32:GD62534],[1-14:u32:404353],8);     // バックアップした IN0～7 を元に戻す
    [1-14:u32:404169] = [u32:GD62554];      // バックアップした JOG 移動量を元に戻す

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD62550] = [1-14:s32:400199];
    [1-14:s32:400909] = [s32:GD62550];

    // NET-IN の入力値をクリアする
    [1-14:w:400126] = 0;

    // Configuration の実行
    [1-14:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD62556] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62556] > 1 && [b:GB62157] == 1){

    [1-14:s32:400395] = 1;          // P-PRESET(プリセット)実行する
    [1-14:s32:400909] = [s32:GD62552]; // プリセット位置パラメータを元に戻す
    [1-14:s32:400397] = 0;          // Configuration 実行のゼロクリア
    [1-14:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62156] = 0;              // 運転終了開始トリガのクリア
                              // (このスクリプトの起動トリガを終了する)
    [b:GB62157] = 0;              // 運転終了中フラグのクリア
}

// 15 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62166] == 1 && [b:GB62167] == 0){

    [b:GB62165] = 0;                    // タッチパネル運転中フラグのクリア
    [b:GB62167] = 1;                    // 運転終了中フラグ(1：運転終了中)
    bmov([u32:GD62602],[1-15:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
    bmov([u32:GD62634],[1-15:u32:404353],8);     // バックアップした IN0～7 を元に戻す
    [1-15:u32:404169] = [u32:GD62654];      // バックアップした JOG 移動量を元に戻す

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD62650] = [1-15:s32:400199];
    [1-15:s32:400909] = [s32:GD62650];

    // NET-IN の入力値をクリアする
    [1-15:w:400126] = 0;

    // Configuration の実行
    [1-15:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD62656] = [w:GS7];
}
```

```
// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62656] > 1 && [b:GB62167] == 1){

   [1-15:s32:400395] = 1;           // P-PRESET(プリセット)実行する
   [1-15:s32:400909] = [s32:GD62652]; // プリセット位置パラメータを元に戻す
   [1-15:s32:400397] = 0;           // Configuration 実行のゼロクリア
   [1-15:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62166] = 0;              // 運転終了開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
   [b:GB62167] = 0;              // 運転終了中フラグのクリア
}

// 16 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62176] == 1 && [b:GB62177] == 0){

   [b:GB62175] = 0;                  // タッチパネル運転中フラグのクリア
   [b:GB62177] = 1;                  // 運転終了中フラグ(1:運転終了中)
   bmov([u32:GD62702],[1-16:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
   bmov([u32:GD62734],[1-16:u32:404353],8);     // バックアップした IN0～7 を元に戻す
   [1-16:u32:404169] = [u32:GD62754];      // バックアップした JOG 移動量を元に戻す

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD62750] = [1-16:s32:400199];
   [1-16:s32:400909] = [s32:GD62750];

   // NET-IN の入力値をクリアする
   [1-16:w:400126] = 0;

   // Configuration の実行
   [1-16:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD62756] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62756] > 1 && [b:GB62177] == 1){

   [1-16:s32:400395] = 1;           // P-PRESET(プリセット)実行する
   [1-16:s32:400909] = [s32:GD62752]; // プリセット位置パラメータを元に戻す
   [1-16:s32:400397] = 0;           // Configuration 実行のゼロクリア
   [1-16:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62176] = 0;              // 運転終了開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
   [b:GB62177] = 0;              // 運転終了中フラグのクリア
}
```

スクリプト No.	31028	スクリプト名	Script31028
コメント	13-16_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62150

ベース画面 31025 のスクリプト No.31028 と同じです。

スクリプト No.	31029	スクリプト名	Script31029

コメント	13-16_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62153

ベース画面 31025 のスクリプト No.31029 と同じです。

スクリプト No.	31030	スクリプト名	Script31030
コメント	13-16_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62155

ベース画面 31025 のスクリプト No.31030 と同じです。

スクリプト No.	31031	スクリプト名	Script31031
コメント	13-16_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62156

ベース画面 31025 のスクリプト No.31031 と同じです。

スクリプト No.	31028	スクリプト名	Script31028
コメント	13-16_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62160

ベース画面 31025 のスクリプト No.31028 と同じです。

スクリプト No.	31029	スクリプト名	Script31029
コメント	13-16_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62163

ベース画面 31025 のスクリプト No.31029 と同じです。

スクリプト No.	31030	スクリプト名	Script31030
コメント	13-16_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62165

ベース画面 31025 のスクリプト No.31030 と同じです。

スクリプト No.	31031	スクリプト名	Script31031
コメント	13-16_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62166

ベース画面 31025 のスクリプト No.31031 と同じです。

スクリプト No.	31028	スクリプト名	Script31028
コメント	13-16_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62170

ベース画面 31025 のスクリプト No.31028 と同じです。

スクリプト No.	31029	スクリプト名	Script31029
コメント	13-16_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62173

ベース画面 31025 のスクリプト No.31029 と同じです。

スクリプト No.	31030	スクリプト名	Script31030
コメント	13-16_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62175

ベース画面 31025 のスクリプト No.31030 と同じです。

スクリプト No.	31031	スクリプト名	Script31031
コメント	13-16_マルチ運転終了		

データ形式	符号付き BIN16	トリガ種別	ON 中 GB62176
ベース画面 31025 のスクリプト No.31031 と同じです。			

ベース画面 31026

スクリプト No.	31032	スクリプト名	Script31032
コメント	17-20_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62180

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 17-20 軸のマルチ運転の画面スクリプト

// 17-20_マルチ運転開始_終了前確認スクリプト No.31032
// 17-20 軸のマルチ運転画面のそれぞれの開始ボタンの ON 中周期、2 秒で、スクリプトを起動

// [b:GB62180] : 17 軸目の起動トリガ、ビットセットで起動
// [b:GB62190] : 18 軸目の起動トリガ、ビットセットで起動
// [b:GB62200] : 19 軸目の起動トリガ、ビットセットで起動
// [b:GB62210] : 20 軸目の起動トリガ、ビットセットで起動
// [b:GB62185] : 17 軸目のタッチパネル運転中フラグ
// [b:GB62195] : 18 軸目のタッチパネル運転中フラグ
// [b:GB62205] : 19 軸目のタッチパネル運転中フラグ
// [b:GB62215] : 20 軸目のタッチパネル運転中フラグ

// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 17 軸目
if([b:GB62180] == 1){
  [w:GD61000] = 17;

  // アラームチェック
  if([1-17:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62181] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62181] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-17:s32:400201] == 0){   // 運転していない場合
    [b:GB62182] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62182] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62181] == 1 && [b:GB62182] == 1){
    if([b:GB62185] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62180] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62183] = 1;    // 運転準備開始トリガ
      [b:GB62181] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62182] = 0;    // 運転中確認フラグのクリア
```

```
    } else {                // タッチパネル運転中なら、運転終了準備をする
      [b:GB62180] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62186] = 1;    // 運転終了開始トリガ
      [b:GB62181] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62182] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62180] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62181] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62182] = 0;    // 運転中確認フラグのクリア
  }
}

// 18 軸目
if([b:GB62190] == 1){
  [w:GD61000] = 18;

  // アラームチェック
  if([1-18:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62191] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62191] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-18:s32:400201] == 0){   // 運転していない場合
    [b:GB62192] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62192] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62191] == 1 && [b:GB62192] == 1){
    if([b:GB62195] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62190] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62193] = 1;    // 運転準備開始トリガ
      [b:GB62191] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62192] = 0;    // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
      [b:GB62190] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62196] = 1;    // 運転終了開始トリガ
      [b:GB62191] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62192] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62190] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62191] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62192] = 0;    // 運転中確認フラグのクリア
  }
}

// 19 軸目
if([b:GB62200] == 1){
```

```
    [w:GD61000] = 19;

    // アラームチェック
    if([1-19:s32:400129] == 0){    // アラームが発生していない場合
        [b:GB62201] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
        [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
    } else {                    // アラームが発生している場合
        [b:GB62201] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
        [w:GD60004] = 32007;        // アラーム発生中確認オーバーラップウィンドウの表示
    }

    // 運転中チェック
    if([1-19:s32:400201] == 0){    // 運転していない場合
        [b:GB62202] = 1;        // 運転中確認フラグ(1:運転していない)
    } else {                    // 運転中の場合
        [b:GB62202] = 0;        // 運転中確認フラグ(0:運転中)
        [w:GD60004] = 32008;        // モーター運転中確認オーバーラップウィンドウの表示
    }

    // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
    if([b:GB62201] == 1 && [b:GB62202] == 1){
        if([b:GB62205] == 0){      // タッチパネル運転中でないなら、運転開始準備をする
            [b:GB62200] = 0;    // このスクリプトの起動トリガを終了する
            [b:GB62203] = 1;    // 運転準備開始トリガ
            [b:GB62201] = 0;    // アラーム発生状態確認フラグのクリア
            [b:GB62202] = 0;    // 運転中確認フラグのクリア
        } else {                // タッチパネル運転中なら、運転終了準備をする
            [b:GB62200] = 0;    // このスクリプトの起動トリガを終了する
            [b:GB62206] = 1;    // 運転終了開始トリガ
            [b:GB62201] = 0;    // アラーム発生状態確認フラグのクリア
            [b:GB62202] = 0;    // 運転中確認フラグのクリア
        }
    } else {
            [b:GB62200] = 0;    // このスクリプトの起動トリガを終了する
            [b:GB62201] = 0;    // アラーム発生状態確認フラグのクリア
            [b:GB62202] = 0;    // 運転中確認フラグのクリア
    }
}

// 20 軸目
if([b:GB62210] == 1){
    [w:GD61000] = 20;

    // アラームチェック
    if([1-20:s32:400129] == 0){    // アラームが発生していない場合
        [b:GB62211] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
        [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
    } else {                    // アラームが発生している場合
        [b:GB62211] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
        [w:GD60004] = 32007;        // アラーム発生中確認オーバーラップウィンドウの表示
    }

    // 運転中チェック
    if([1-20:s32:400201] == 0){    // 運転していない場合
        [b:GB62212] = 1;        // 運転中確認フラグ(1:運転していない)
    } else {                    // 運転中の場合
```

```
    [b:GB62212] = 0;      // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62211] == 1 && [b:GB62212] == 1){
    if([b:GB62215] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62210] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62213] = 1;    // 運転準備開始トリガ
      [b:GB62211] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62212] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62210] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62216] = 1;    // 運転終了開始トリガ
      [b:GB62211] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62212] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62210] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62211] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62212] = 0;    // 運転中確認フラグのクリア
  }
}
```

スクリプトNo.	31033	スクリプト名	Script31033
コメント	17-20_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62183

```
// タッチパネルから運転できるように準備するスクリプト
// NET-INなどの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 17-20軸マルチ運転の画面スクリプト

// 17-20_マルチ運転準備スクリプト No.31033
// 17-20_マルチ運転開始_終了前確認スクリプト(No.31032)の GB62183、GB62193、GB62203、GB62213 の ON
中で、スクリプトを起動

// [b:GB62183],    [b:GB62193],    [b:GB62203],    [b:GB62213]   : 運転準備開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62184],    [b:GB62194],    [b:GB62204],    [b:GB62214]   : 運転準備中フラグ
// [b:GB62185],    [b:GB62195],    [b:GB62205],    [b:GB62215]   : タッチパネル運転中フラグ
// [u32:GD62802], [u32:GD62902], [u32:GD63002], [u32:GD63102] : NET-IN0～15のバックアップデバイ
ス
// [u32:GD62834], [u32:GD62934], [u32:GD63034], [u32:GD63134] : IN0～7のバックアップデバイス
// [s32:GD62850], [s32:GD62950], [s32:GD63050], [s32:GD63150] : モニタの指令位置のバックアップ
デバイス
// [s32:GD62852], [s32:GD62952], [s32:GD63052], [u32:GD63152] : プリセット位置のバックアップデ
バイス
// [u32:GD62854], [u32:GD62954], [u32:GD63054], [u32:GD63154] : JOG移動量のバックアップデバイス
(GOTの最小移動量に使用)
// [w:GD62856],   [w:GD62956],    [w:GD63056],   [w:GD63156]     : Configuration 実行待ち時間用
タイマー

// 17軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62183] == 1 && [b:GB62184] == 0){

  [b:GB62184] = 1;                      // 運転準備中フラグ(1：運転準備中)
  bmov([1-17:u32:404449],[u32:GD62802],16);    // NET-IN0～15のバックアップ
```

```
   bmov([1-17:u32:404353],[u32:GD62834],8);     // IN0～7のバックアップ
   [s32:GD62852] = [1-17:s32:400909];      // プリセット位置のバックアップ
   [u32:GD62854] = [1-17:u32:404169];      // JOG移動量のバックアップ(GOTの最小移動量に使用)

   // NET-INの入力値をクリアする
   [1-17:w:400126] = 0;

   // タッチパネルから運転する為に信号の割当を変更する
   [1-17:u32:404449] = 48;     // NET-IN0  M0
   [1-17:u32:404451] = 49;     // NET-IN1  M1
   [1-17:u32:404453] = 50;     // NET-IN2  M2
   [1-17:u32:404455] = 51;     // NET-IN3  M3
   [1-17:u32:404457] = 52;     // NET-IN4  M4
   [1-17:u32:404459] = 53;     // NET-IN5  M5
   [1-17:u32:404461] = 4;   // NET-IN6  START
   [1-17:u32:404463] = 3;   // NET-IN7  HOME
   [1-17:u32:404465] = 18;     // NET-IN8  STOP
   [1-17:u32:404467] = 9;   // NET-IN9  MS1
   [1-17:u32:404469] = 10;     // NET-IN10 MS2
   [1-17:u32:404471] = 5;   // NET-IN11 SSTART
   [1-17:u32:404473] = 6;   // NET-IN12 +JOG
   [1-17:u32:404475] = 7;   // NET-IN13 -JOG
   [1-17:u32:404477] = 1;   // NET-IN14 FWD
   [1-17:u32:404479] = 2;   // NET-IN15 RVS

   [1-17:u32:404353] = 32;  // IN0 R0
   [1-17:u32:404355] = 33;  // IN1 R1
   [1-17:u32:404357] = 34;  // IN2 R2
   [1-17:u32:404359] = 35;  // IN3 R3
   [1-17:u32:404361] = 36;  // IN4 R4
   [1-17:u32:404363] = 37;  // IN5 R5
   [1-17:u32:404365] = 18;  // IN6 STOP
   [1-17:u32:404367] = 39;  // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD62850] = [1-17:s32:400199];
   [1-17:s32:400909] = [s32:GD62850];

   // Configurationの実行
   [1-17:s32:400397] = 1;

   // Configuration実行待ち時間のタイマー記録(単位は秒)
   [w:GD62856] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62856] > 1 && [b:GB62184] == 1){

   [1-17:s32:400395] = 1;          // P-PRESET(プリセット)実行する
   [1-17:s32:400909] = [s32:GD62852]; // バックアップしたプリセット位置パラメータを元に戻す
   [1-17:s32:400397] = 0;          // Configuration実行のゼロクリア
   [1-17:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62183] = 0;              // 運転準備開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
   [b:GB62184] = 0;              // 運転準備中フラグのクリア
```

```
    [b:GB62185] = 1;              // タッチパネル運転中フラグ(1：運転中)
}

// 18 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62193] == 1 && [b:GB62194] == 0){

    [b:GB62194] = 1;                    // 運転準備中フラグ(1：運転準備中)
    bmov([1-18:u32:404449],[u32:GD62902],16);    // NET-IN0～15 のバックアップ
    bmov([1-18:u32:404353],[u32:GD62934],8);     // IN0～7 のバックアップ
    [s32:GD62952] = [1-18:s32:400909];       // プリセット位置のバックアップ
    [u32:GD62954] = [1-18:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

    // NET-IN の入力値をクリアする
    [1-18:w:400126] = 0;

    // タッチパネルから運転する為に信号の割当を変更する
    [1-18:u32:404449] = 48;     // NET-IN0  M0
    [1-18:u32:404451] = 49;     // NET-IN1  M1
    [1-18:u32:404453] = 50;     // NET-IN2  M2
    [1-18:u32:404455] = 51;     // NET-IN3  M3
    [1-18:u32:404457] = 52;     // NET-IN4  M4
    [1-18:u32:404459] = 53;     // NET-IN5  M5
    [1-18:u32:404461] = 4;   // NET-IN6  START
    [1-18:u32:404463] = 3;   // NET-IN7  HOME
    [1-18:u32:404465] = 18;     // NET-IN8  STOP
    [1-18:u32:404467] = 9;   // NET-IN9  MS1
    [1-18:u32:404469] = 10;     // NET-IN10 MS2
    [1-18:u32:404471] = 5;   // NET-IN11 SSTART
    [1-18:u32:404473] = 6;   // NET-IN12 +JOG
    [1-18:u32:404475] = 7;   // NET-IN13 -JOG
    [1-18:u32:404477] = 1;   // NET-IN14 FWD
    [1-18:u32:404479] = 2;   // NET-IN15 RVS

    [1-18:u32:404353] = 32;  // IN0 R0
    [1-18:u32:404355] = 33;  // IN1 R1
    [1-18:u32:404357] = 34;  // IN2 R2
    [1-18:u32:404359] = 35;  // IN3 R3
    [1-18:u32:404361] = 36;  // IN4 R4
    [1-18:u32:404363] = 37;  // IN5 R5
    [1-18:u32:404365] = 18;  // IN6 STOP
    [1-18:u32:404367] = 39;  // IN7 R7

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD62950] = [1-18:s32:400199];
    [1-18:s32:400909] = [s32:GD62950];

    // Configuration の実行
    [1-18:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD62956] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD62956] > 1 && [b:GB62194] == 1){
```

```
  [1-18:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-18:s32:400909] = [s32:GD62952]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-18:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-18:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62193] = 0;              // 運転準備開始トリガのクリア
                              //(このスクリプトの起動トリガを終了する)
  [b:GB62194] = 0;              // 運転準備中フラグのクリア
  [b:GB62195] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 19 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62203] == 1 && [b:GB62204] == 0){

  [b:GB62204] = 1;                    // 運転準備中フラグ(1:運転準備中)
  bmov([1-19:u32:404449],[u32:GD63002],16);    // NET-IN0～15 のバックアップ
  bmov([1-19:u32:404353],[u32:GD63034],8);     // IN0～7 のバックアップ
  [s32:GD63052] = [1-19:s32:400909];      // プリセット位置のバックアップ
  [u32:GD63054] = [1-19:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-19:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-19:u32:404449] = 48;     // NET-IN0  M0
  [1-19:u32:404451] = 49;     // NET-IN1  M1
  [1-19:u32:404453] = 50;     // NET-IN2  M2
  [1-19:u32:404455] = 51;     // NET-IN3  M3
  [1-19:u32:404457] = 52;     // NET-IN4  M4
  [1-19:u32:404459] = 53;     // NET-IN5  M5
  [1-19:u32:404461] = 4;   // NET-IN6  START
  [1-19:u32:404463] = 3;   // NET-IN7  HOME
  [1-19:u32:404465] = 18;     // NET-IN8  STOP
  [1-19:u32:404467] = 9;   // NET-IN9  MS1
  [1-19:u32:404469] = 10;     // NET-IN10 MS2
  [1-19:u32:404471] = 5;   // NET-IN11 SSTART
  [1-19:u32:404473] = 6;   // NET-IN12 +JOG
  [1-19:u32:404475] = 7;   // NET-IN13 -JOG
  [1-19:u32:404477] = 1;   // NET-IN14 FWD
  [1-19:u32:404479] = 2;   // NET-IN15 RVS

  [1-19:u32:404353] = 32;  // IN0 R0
  [1-19:u32:404355] = 33;  // IN1 R1
  [1-19:u32:404357] = 34;  // IN2 R2
  [1-19:u32:404359] = 35;  // IN3 R3
  [1-19:u32:404361] = 36;  // IN4 R4
  [1-19:u32:404363] = 37;  // IN5 R5
  [1-19:u32:404365] = 18;  // IN6 STOP
  [1-19:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63050] = [1-19:s32:400199];
  [1-19:s32:400909] = [s32:GD63050];
```

```
  // Configurationの実行
  [1-19:s32:400397] = 1;

  // Configuration実行待ち時間のタイマー記録(単位は秒)
  [w:GD63056] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63056] > 1 && [b:GB62204] == 1){

  [1-19:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-19:s32:400909] = [s32:GD63052]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-19:s32:400397] = 0;          // Configuration実行のゼロクリア
  [1-19:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62203] = 0;              // 運転準備開始トリガのクリア
                            // (このスクリプトの起動トリガを終了する)
  [b:GB62204] = 0;              // 運転準備中フラグのクリア
  [b:GB62205] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 20軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62213] == 1 && [b:GB62214] == 0){

  [b:GB62214] = 1;                    // 運転準備中フラグ(1:運転準備中)
  bmov([1-20:u32:404449],[u32:GD63102],16);    // NET-IN0～15のバックアップ
  bmov([1-20:u32:404353],[u32:GD63134],8);     // IN0～7のバックアップ
  [s32:GD63152] = [1-20:s32:400909];       // プリセット位置のバックアップ
  [u32:GD63154] = [1-20:u32:404169];       // JOG移動量のバックアップ(GOTの最小移動量に使用)

  // NET-INの入力値をクリアする
  [1-20:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-20:u32:404449] = 48;     // NET-IN0  M0
  [1-20:u32:404451] = 49;     // NET-IN1  M1
  [1-20:u32:404453] = 50;     // NET-IN2  M2
  [1-20:u32:404455] = 51;     // NET-IN3  M3
  [1-20:u32:404457] = 52;     // NET-IN4  M4
  [1-20:u32:404459] = 53;     // NET-IN5  M5
  [1-20:u32:404461] = 4;   // NET-IN6  START
  [1-20:u32:404463] = 3;   // NET-IN7  HOME
  [1-20:u32:404465] = 18;     // NET-IN8  STOP
  [1-20:u32:404467] = 9;   // NET-IN9  MS1
  [1-20:u32:404469] = 10;     // NET-IN10 MS2
  [1-20:u32:404471] = 5;   // NET-IN11 SSTART
  [1-20:u32:404473] = 6;   // NET-IN12 +JOG
  [1-20:u32:404475] = 7;   // NET-IN13 -JOG
  [1-20:u32:404477] = 1;   // NET-IN14 FWD
  [1-20:u32:404479] = 2;   // NET-IN15 RVS

  [1-20:u32:404353] = 32;  // IN0 R0
  [1-20:u32:404355] = 33;  // IN1 R1
  [1-20:u32:404357] = 34;  // IN2 R2
  [1-20:u32:404359] = 35;  // IN3 R3
```

```
  [1-20:u32:404361] = 36;  // IN4 R4
  [1-20:u32:404363] = 37;  // IN5 R5
  [1-20:u32:404365] = 18;  // IN6 STOP
  [1-20:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63150] = [1-20:s32:400199];
  [1-20:s32:400909] = [s32:GD63150];

  // Configurationの実行
  [1-20:s32:400397] = 1;

  // Configuration実行待ち時間のタイマー記録(単位は秒)
  [w:GD63156] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63156] > 1 && [b:GB62214] == 1){

  [1-20:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-20:s32:400909] = [s32:GD63152]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-20:s32:400397] = 0;          // Configuration実行のゼロクリア
  [1-20:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62213] = 0;             // 運転準備開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
  [b:GB62214] = 0;             // 運転準備中フラグのクリア
  [b:GB62215] = 1;             // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプトNo.	31034	スクリプト名	Script31034
コメント	17-20_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62185

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データNo.をM0～M5(割付直したNET-IN)に反映させる
// 反映したM0～M5は、ドライバ入力指令(400126)にて、ONする

// 17-20_マルチ運転中スクリプト No.31034
// 17-20_マルチ運転準備スクリプト(No.31033)のGB62185、GB62195、GB62205、GB62215のON中で、スクリ
プトを起動

// [b:GB62185], [b:GB62195], [b:GB62205], [b:GB62215] : タッチパネル運転中フラグ(このスクリプト
の起動トリガ)
// また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-17:u32:400195] ～ [1-20:u32:400195] : モニタの現在の選択データNo.のModbusアドレス
// [u32:GD62800], [u32GD62900], [GD63000], [u32:GD63100] : 運転データNo.のデバイス

// 選択した運転データNo.をドライバ入力指令のM0～M5に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データNo.(400195)と選択した運転データNo.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

// 17軸目
if([b:GB62185] == 1){

  if([1-17:u32:400195] != [u32:GD62800]){
```

```
      [1-17:b:400126.b0] = [b:GD62800.b0];    // M0
      [1-17:b:400126.b1] = [b:GD62800.b1];    // M1
      [1-17:b:400126.b2] = [b:GD62800.b2];    // M2
      [1-17:b:400126.b3] = [b:GD62800.b3];    // M3
      [1-17:b:400126.b4] = [b:GD62800.b4];    // M4
      [1-17:b:400126.b5] = [b:GD62800.b5];    // M5
   }
}

// 18 軸目
if([b:GB62195] == 1){

   if([1-18:u32:400195] != [u32:GD62900]){
      [1-18:b:400126.b0] = [b:GD62900.b0];    // M0
      [1-18:b:400126.b1] = [b:GD62900.b1];    // M1
      [1-18:b:400126.b2] = [b:GD62900.b2];    // M2
      [1-18:b:400126.b3] = [b:GD62900.b3];    // M3
      [1-18:b:400126.b4] = [b:GD62900.b4];    // M4
      [1-18:b:400126.b5] = [b:GD62900.b5];    // M5
   }
}

// 19 軸目
if([b:GB62205] == 1){

   if([1-19:u32:400195] != [u32:GD63000]){
      [1-19:b:400126.b0] = [b:GD63000.b0];    // M0
      [1-19:b:400126.b1] = [b:GD63000.b1];    // M1
      [1-19:b:400126.b2] = [b:GD63000.b2];    // M2
      [1-19:b:400126.b3] = [b:GD63000.b3];    // M3
      [1-19:b:400126.b4] = [b:GD63000.b4];    // M4
      [1-19:b:400126.b5] = [b:GD63000.b5];    // M5
   }
}

// 20 軸目
if([b:GB62215] == 1){

   if([1-20:u32:400195] != [u32:GD63100]){
      [1-20:b:400126.b0] = [b:GD63100.b0];    // M0
      [1-20:b:400126.b1] = [b:GD63100.b1];    // M1
      [1-20:b:400126.b2] = [b:GD63100.b2];    // M2
      [1-20:b:400126.b3] = [b:GD63100.b3];    // M3
      [1-20:b:400126.b4] = [b:GD63100.b4];    // M4
      [1-20:b:400126.b5] = [b:GD63100.b5];    // M5
   }
}
```

スクリプト No.	31035	スクリプト名	Script31035
コメント	17-20_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62186

```
// タッチパネルからの運転を終了するスクリプト
// バックアップした NET-IN などの信号を元に戻したり、必要なパラメータなどを元に戻す
// 17-20 軸のマルチ運転の画面スクリプト

// 17-20_マルチ運転終了スクリプト No.31035
// 17-20_マルチ運転開始_終了前確認スクリプト(No.31032)の GB62186、GB62196、GB62206、GB62216 の ON
```

```
中で、スクリプトを起動

// [b:GB62185],    [b:GB62195],    [b:GB62205],    [b:GB62215]   : タッチパネル運転中フラグ
// [b:GB62186],    [b:GB62196],    [b:GB62206],    [b:GB62216]   : 運転終了開始トリガ(このスクリプトの起動トリガ)
// [b:GB62187],    [b:GB62197],    [b:GB62207],    [b:GB62217]   : 運転終了中フラグ
// [u32:GD62802], [u32:GD62902], [u32:GD63002], [u32:GD63102] : NET-IN0～15 のバックアップデバイス
// [u32:GD62834], [u32:GD62934], [u32:GD63034], [u32:GD63134] : IN0～7 のバックアップデバイス
// [s32:GD62850], [s32:GD62950], [s32:GD63050], [s32:GD63150] : モニタの指令位置のバックアップデバイス
// [s32:GD62852], [s32:GD62952], [s32:GD63052], [s32:GD63152] : プリセット位置のバックアップデバイス
// [u32:GD62854], [u32:GD62954], [u32:GD63054], [u32:GD63154] : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD62856],   [w:GD62956],   [w:GD63056],   [w:GD63156]     : Configuration 実行待ち時間用タイマー

// 17 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62186] == 1 && [b:GB62187] == 0){

  [b:GB62185] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62187] = 1;                    // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD62802],[1-17:u32:404449],16);   // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD62834],[1-17:u32:404353],8);    // バックアップした IN0～7 を元に戻す
  [1-17:u32:404169] = [u32:GD62854];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62850] = [1-17:s32:400199];
  [1-17:s32:400909] = [s32:GD62850];

  // NET-IN の入力値をクリアする
  [1-17:w:400126] = 0;

  // Configuration の実行
  [1-17:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD62856] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62856] > 1 && [b:GB62187] == 1){

  [1-17:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-17:s32:400909] = [s32:GD62852]; // プリセット位置パラメータを元に戻す
  [1-17:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-17:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62186] = 0;              // 運転終了開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62187] = 0;              // 運転終了中フラグのクリア
}

// 18 軸目
```

```
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62196] == 1 && [b:GB62197] == 0){

  [b:GB62195] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62197] = 1;                    // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD62902],[1-18:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD62934],[1-18:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-18:u32:404169] = [u32:GD62954];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD62950] = [1-18:s32:400199];
  [1-18:s32:400909] = [s32:GD62950];

  // NET-IN の入力値をクリアする
  [1-18:w:400126] = 0;

  // Configuration の実行
  [1-18:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD62956] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD62956] > 1 && [b:GB62197] == 1){

  [1-18:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-18:s32:400909] = [s32:GD62952]; // プリセット位置パラメータを元に戻す
  [1-18:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-18:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62196] = 0;               // 運転終了開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62197] = 0;               // 運転終了中フラグのクリア
}

// 19 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62206] == 1 && [b:GB62207] == 0){

  [b:GB62205] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62207] = 1;                    // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63002],[1-19:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63034],[1-19:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-19:u32:404169] = [u32:GD63054];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63050] = [1-19:s32:400199];
  [1-19:s32:400909] = [s32:GD63050];

  // NET-IN の入力値をクリアする
  [1-19:w:400126] = 0;

  // Configuration の実行
  [1-19:s32:400397] = 1;
```

```
  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63056] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63056] > 1 && [b:GB62207] == 1){

  [1-19:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-19:s32:400909] = [s32:GD63052]; // プリセット位置パラメータを元に戻す
  [1-19:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-19:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62206] = 0;             // 運転終了開始トリガのクリア
                        // (このスクリプトの起動トリガを終了する)
  [b:GB62207] = 0;             // 運転終了中フラグのクリア
}

// 20 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62216] == 1 && [b:GB62217] == 0){

  [b:GB62215] = 0;                // タッチパネル運転中フラグのクリア
  [b:GB62217] = 1;                // 運転終了中フラグ(1 : 運転終了中)
  bmov([u32:GD63102],[1-20:u32:404449],16);   // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63134],[1-20:u32:404353],8);    // バックアップした IN0～7 を元に戻す
  [1-20:u32:404169] = [u32:GD63154];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63150] = [1-20:s32:400199];
  [1-20:s32:400909] = [s32:GD63150];

  // NET-IN の入力値をクリアする
  [1-20:w:400126] = 0;

  // Configuration の実行
  [1-20:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63156] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63156] > 1 && [b:GB62217] == 1){

  [1-20:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-20:s32:400909] = [s32:GD63152]; // プリセット位置パラメータを元に戻す
  [1-20:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-20:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62216] = 0;             // 運転終了開始トリガのクリア
                        // (このスクリプトの起動トリガを終了する)
  [b:GB62217] = 0;             // 運転終了中フラグのクリア
}
```

スクリプト No.	31032	スクリプト名	Script31032
コメント	17-20_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62190

ベース画面31026のスクリプトNo.31032と同じです。			

スクリプトNo.	31033	スクリプト名	Script31033
コメント	17-20_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62193
ベース画面31026のスクリプトNo.31033と同じです。			

スクリプトNo.	31034	スクリプト名	Script31034
コメント	17-20_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62195
ベース画面31026のスクリプトNo.31034と同じです。			

スクリプトNo.	31035	スクリプト名	Script31035
コメント	17-20_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62196
ベース画面31026のスクリプトNo.31035と同じです。			

スクリプトNo.	31032	スクリプト名	Script31032
コメント	17-20_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62200
ベース画面31026のスクリプトNo.31032と同じです。			

スクリプトNo.	31033	スクリプト名	Script31033
コメント	17-20_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62203
ベース画面31026のスクリプトNo.31033と同じです。			

スクリプトNo.	31034	スクリプト名	Script31034
コメント	17-20_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62205
ベース画面31026のスクリプトNo.31034と同じです。			

スクリプトNo.	31035	スクリプト名	Script31035
コメント	17-20_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62206
ベース画面31026のスクリプトNo.31035と同じです。			

スクリプトNo.	31032	スクリプト名	Script31032
コメント	17-20_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62210
ベース画面31026のスクリプトNo.31032と同じです。			

スクリプトNo.	31033	スクリプト名	Script31033
コメント	17-20_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62213
ベース画面31026のスクリプトNo.31033と同じです。			

スクリプトNo.	31034	スクリプト名	Script31034
コメント	17-20_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62215

ベース画面 31026 のスクリプト No.31034 と同じです。			
スクリプト No.	31035	スクリプト名	Script31035
コメント	17-20_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62216
ベース画面 31026 のスクリプト No.31035 と同じです。			

ベース画面 31027

スクリプト No.	31036	スクリプト名	Script31036
コメント	21-24_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62220

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 21-24 軸のマルチ運転の画面スクリプト

// 21-24_マルチ運転開始_終了前確認スクリプト No.31036
// 21-24 軸のマルチ運転画面のそれぞれの開始ボタンの ON 中周期、2 秒で、スクリプトを起動

// [b:GB62220] : 21 軸目の起動トリガ、ビットセットで起動
// [b:GB62230] : 22 軸目の起動トリガ、ビットセットで起動
// [b:GB62240] : 23 軸目の起動トリガ、ビットセットで起動
// [b:GB62250] : 24 軸目の起動トリガ、ビットセットで起動
// [b:GB62225] : 21 軸目のタッチパネル運転中フラグ
// [b:GB62235] : 22 軸目のタッチパネル運転中フラグ
// [b:GB62245] : 23 軸目のタッチパネル運転中フラグ
// [b:GB62255] : 24 軸目のタッチパネル運転中フラグ

// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 21 軸目
if([b:GB62220] == 1){
  [w:GD61000] = 21;

  // アラームチェック
  if([1-21:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62221] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62221] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-21:s32:400201] == 0){   // 運転していない場合
    [b:GB62222] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62222] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62221] == 1 && [b:GB62222] == 1){
```

```
      if([b:GB62225] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
         [b:GB62220] = 0;     // このスクリプトの起動トリガを終了する
         [b:GB62223] = 1;     // 運転準備開始トリガ
         [b:GB62221] = 0;     // アラーム発生状態確認フラグのクリア
         [b:GB62222] = 0;     // 運転中確認フラグのクリア
      } else {                // タッチパネル運転中なら、運転終了準備をする
         [b:GB62220] = 0;     // このスクリプトの起動トリガを終了する
         [b:GB62226] = 1;     // 運転終了開始トリガ
         [b:GB62221] = 0;     // アラーム発生状態確認フラグのクリア
         [b:GB62222] = 0;     // 運転中確認フラグのクリア
      }
   } else {
         [b:GB62220] = 0;     // このスクリプトの起動トリガを終了する
         [b:GB62221] = 0;     // アラーム発生状態確認フラグのクリア
         [b:GB62222] = 0;     // 運転中確認フラグのクリア
   }
}

// 22 軸目
if([b:GB62230] == 1){
   [w:GD61000] = 22;

   // アラームチェック
   if([1-22:s32:400129] == 0){    // アラームが発生していない場合
      [b:GB62231] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
      [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
   } else {                   // アラームが発生している場合
      [b:GB62231] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
      [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
   }

   // 運転中チェック
   if([1-22:s32:400201] == 0){    // 運転していない場合
      [b:GB62232] = 1;        // 運転中確認フラグ(1:運転していない)
   } else {                   // 運転中の場合
      [b:GB62232] = 0;        // 運転中確認フラグ(0:運転中)
      [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
   }

   // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
   if([b:GB62231] == 1 && [b:GB62232] == 1){
      if([b:GB62235] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
         [b:GB62230] = 0;     // このスクリプトの起動トリガを終了する
         [b:GB62233] = 1;     // 運転準備開始トリガ
         [b:GB62231] = 0;     // アラーム発生状態確認フラグのクリア
         [b:GB62232] = 0;     // 運転中確認フラグのクリア
      } else {                // タッチパネル運転中なら、運転終了準備をする
         [b:GB62230] = 0;     // このスクリプトの起動トリガを終了する
         [b:GB62236] = 1;     // 運転終了開始トリガ
         [b:GB62231] = 0;     // アラーム発生状態確認フラグのクリア
         [b:GB62232] = 0;     // 運転中確認フラグのクリア
      }
   } else {
         [b:GB62230] = 0;     // このスクリプトの起動トリガを終了する
         [b:GB62231] = 0;     // アラーム発生状態確認フラグのクリア
         [b:GB62232] = 0;     // 運転中確認フラグのクリア
```

```
    }
}

// 23 軸目
if([b:GB62240] == 1){
    [w:GD61000] = 23;

    // アラームチェック
    if([1-23:s32:400129] == 0){     // アラームが発生していない場合
        [b:GB62241] = 1;            // アラーム発生状態確認フラグ(1:アラーム発生なし)
        [w:GD60004] = 0;            // アラーム発生中確認オーバーラップウィンドウの消去
    } else {                        // アラームが発生している場合
        [b:GB62241] = 0;            // アラーム発生状態確認フラグ(0:アラーム発生中)
        [w:GD60004] = 32007;        // アラーム発生中確認オーバーラップウィンドウの表示
    }

    // 運転中チェック
    if([1-23:s32:400201] == 0){     // 運転していない場合
        [b:GB62242] = 1;            // 運転中確認フラグ(1:運転していない)
    } else {                        // 運転中の場合
        [b:GB62242] = 0;            // 運転中確認フラグ(0:運転中)
        [w:GD60004] = 32008;        // モーター運転中確認オーバーラップウィンドウの表示
    }

    // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
    if([b:GB62241] == 1 && [b:GB62242] == 1){
        if([b:GB62245] == 0){       // タッチパネル運転中でないなら、運転開始準備をする
            [b:GB62240] = 0;        // このスクリプトの起動トリガを終了する
            [b:GB62243] = 1;        // 運転準備開始トリガ
            [b:GB62241] = 0;        // アラーム発生状態確認フラグのクリア
            [b:GB62242] = 0;        // 運転中確認フラグのクリア
        } else {                    // タッチパネル運転中なら、運転終了準備をする
            [b:GB62240] = 0;        // このスクリプトの起動トリガを終了する
            [b:GB62246] = 1;        // 運転終了開始トリガ
            [b:GB62241] = 0;        // アラーム発生状態確認フラグのクリア
            [b:GB62242] = 0;        // 運転中確認フラグのクリア
        }
    } else {
            [b:GB62240] = 0;        // このスクリプトの起動トリガを終了する
            [b:GB62241] = 0;        // アラーム発生状態確認フラグのクリア
            [b:GB62242] = 0;        // 運転中確認フラグのクリア
    }
}

// 24 軸目
if([b:GB62250] == 1){
    [w:GD61000] = 24;

    // アラームチェック
    if([1-24:s32:400129] == 0){     // アラームが発生していない場合
        [b:GB62251] = 1;            // アラーム発生状態確認フラグ(1:アラーム発生なし)
        [w:GD60004] = 0;            // アラーム発生中確認オーバーラップウィンドウの消去
    } else {                        // アラームが発生している場合
        [b:GB62251] = 0;            // アラーム発生状態確認フラグ(0:アラーム発生中)
        [w:GD60004] = 32007;        // アラーム発生中確認オーバーラップウィンドウの表示
    }
```

```
  // 運転中チェック
  if([1-24:s32:400201] == 0){   // 運転していない場合
    [b:GB62252] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62252] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62251] == 1 && [b:GB62252] == 1){
    if([b:GB62255] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62250] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62253] = 1;    // 運転準備開始トリガ
      [b:GB62251] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62252] = 0;    // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
      [b:GB62250] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62256] = 1;    // 運転終了開始トリガ
      [b:GB62251] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62252] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62250] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62251] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62252] = 0;    // 運転中確認フラグのクリア
  }
}
```

スクリプト No.	31037	スクリプト名	Script31037
コメント	21-24_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62223

```
// タッチパネルから運転できるように準備するスクリプト
// NET-IN などの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 21-24 軸マルチ運転の画面スクリプト

// 21-24_マルチ運転準備スクリプト No.31037
// 21-24_マルチ運転開始_終了前確認スクリプト(No.31036)の GB62223、GB62233、GB62243、GB62253 の ON
中で、スクリプトを起動

// [b:GB62223],    [b:GB62233],    [b:GB62243],    [b:GB62253]   : 運転準備開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62224],    [b:GB62234],    [b:GB62244],    [b:GB62254]   : 運転準備中フラグ
// [b:GB62225],    [b:GB62235],    [b:GB62245],    [b:GB62255]   : タッチパネル運転中フラグ
// [u32:GD63202], [u32:GD63302], [u32:GD63402], [u32:GD63502] : NET-IN0～15 のバックアップデバイ
ス
// [u32:GD63234], [u32:GD63334], [u32:GD63434], [u32:GD63534]  : IN0～7 のバックアップデバイス
// [s32:GD63250], [s32:GD63350], [s32:GD63450], [s32:GD63550]  : モニタの指令位置のバックアップ
デバイス
// [s32:GD63252], [s32:GD63352], [s32:GD63452], [u32:GD63552]  : プリセット位置のバックアップデ
バイス
// [u32:GD63254], [u32:GD63354], [u32:GD63454], [u32:GD63554] : JOG 移動量のバックアップデバイス
(GOT の最小移動量に使用)
// [w:GD63256],   [w:GD63356],    [w:GD63456],   [w:GD63556]     : Configuration 実行待ち時間用
タイマー

// 21 軸目
```

```
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62223] == 1 && [b:GB62224] == 0){

   [b:GB62224] = 1;                    // 運転準備中フラグ(1：運転準備中)
   bmov([1-21:u32:404449],[u32:GD63202],16);    // NET-IN0～15 のバックアップ
   bmov([1-21:u32:404353],[u32:GD63234],8);     // IN0～7 のバックアップ
   [s32:GD63252] = [1-21:s32:400909];       // プリセット位置のバックアップ
   [u32:GD63254] = [1-21:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

   // NET-IN の入力値をクリアする
   [1-21:w:400126] = 0;

   // タッチパネルから運転する為に信号の割当を変更する
   [1-21:u32:404449] = 48;     // NET-IN0  M0
   [1-21:u32:404451] = 49;     // NET-IN1  M1
   [1-21:u32:404453] = 50;     // NET-IN2  M2
   [1-21:u32:404455] = 51;     // NET-IN3  M3
   [1-21:u32:404457] = 52;     // NET-IN4  M4
   [1-21:u32:404459] = 53;     // NET-IN5  M5
   [1-21:u32:404461] = 4;   // NET-IN6  START
   [1-21:u32:404463] = 3;   // NET-IN7  HOME
   [1-21:u32:404465] = 18;     // NET-IN8  STOP
   [1-21:u32:404467] = 9;   // NET-IN9  MS1
   [1-21:u32:404469] = 10;     // NET-IN10 MS2
   [1-21:u32:404471] = 5;   // NET-IN11 SSTART
   [1-21:u32:404473] = 6;   // NET-IN12 +JOG
   [1-21:u32:404475] = 7;   // NET-IN13 -JOG
   [1-21:u32:404477] = 1;   // NET-IN14 FWD
   [1-21:u32:404479] = 2;   // NET-IN15 RVS

   [1-21:u32:404353] = 32;  // IN0 R0
   [1-21:u32:404355] = 33;  // IN1 R1
   [1-21:u32:404357] = 34;  // IN2 R2
   [1-21:u32:404359] = 35;  // IN3 R3
   [1-21:u32:404361] = 36;  // IN4 R4
   [1-21:u32:404363] = 37;  // IN5 R5
   [1-21:u32:404365] = 18;  // IN6 STOP
   [1-21:u32:404367] = 39;  // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD63250] = [1-21:s32:400199];
   [1-21:s32:400909] = [s32:GD63250];

   // Configuration の実行
   [1-21:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD63256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63256] > 1 && [b:GB62224] == 1){

   [1-21:s32:400395] = 1;           // P-PRESET(プリセット)実行する
   [1-21:s32:400909] = [s32:GD63252]; // バックアップしたプリセット位置パラメータを元に戻す
   [1-21:s32:400397] = 0;           // Configuration 実行のゼロクリア
```

```
  [1-21:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62223] = 0;              // 運転準備開始トリガのクリア
                        // (このスクリプトの起動トリガを終了する)
  [b:GB62224] = 0;              // 運転準備中フラグのクリア
  [b:GB62225] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 22 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62233] == 1 && [b:GB62234] == 0){

  [b:GB62234] = 1;                    // 運転準備中フラグ(1:運転準備中)
  bmov([1-22:u32:404449],[u32:GD63302],16);    // NET-IN0～15 のバックアップ
  bmov([1-22:u32:404353],[u32:GD63334],8);     // IN0～7 のバックアップ
  [s32:GD63352] = [1-22:s32:400909];      // プリセット位置のバックアップ
  [u32:GD63354] = [1-22:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-22:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-22:u32:404449] = 48;     // NET-IN0  M0
  [1-22:u32:404451] = 49;     // NET-IN1  M1
  [1-22:u32:404453] = 50;     // NET-IN2  M2
  [1-22:u32:404455] = 51;     // NET-IN3  M3
  [1-22:u32:404457] = 52;     // NET-IN4  M4
  [1-22:u32:404459] = 53;     // NET-IN5  M5
  [1-22:u32:404461] = 4;   // NET-IN6  START
  [1-22:u32:404463] = 3;   // NET-IN7  HOME
  [1-22:u32:404465] = 18;     // NET-IN8  STOP
  [1-22:u32:404467] = 9;   // NET-IN9  MS1
  [1-22:u32:404469] = 10;     // NET-IN10 MS2
  [1-22:u32:404471] = 5;   // NET-IN11 SSTART
  [1-22:u32:404473] = 6;   // NET-IN12 +JOG
  [1-22:u32:404475] = 7;   // NET-IN13 -JOG
  [1-22:u32:404477] = 1;   // NET-IN14 FWD
  [1-22:u32:404479] = 2;   // NET-IN15 RVS

  [1-22:u32:404353] = 32;  // IN0 R0
  [1-22:u32:404355] = 33;  // IN1 R1
  [1-22:u32:404357] = 34;  // IN2 R2
  [1-22:u32:404359] = 35;  // IN3 R3
  [1-22:u32:404361] = 36;  // IN4 R4
  [1-22:u32:404363] = 37;  // IN5 R5
  [1-22:u32:404365] = 18;  // IN6 STOP
  [1-22:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63350] = [1-22:s32:400199];
  [1-22:s32:400909] = [s32:GD63350];

  // Configuration の実行
  [1-22:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
```

```
    [w:GD63356] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63356] > 1 && [b:GB62234] == 1){

    [1-22:s32:400395] = 1;           // P-PRESET(プリセット)実行する
    [1-22:s32:400909] = [s32:GD63352]; // バックアップしたプリセット位置パラメータを元に戻す
    [1-22:s32:400397] = 0;           // Configuration 実行のゼロクリア
    [1-22:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62233] = 0;                // 運転準備開始トリガのクリア
                                 // (このスクリプトの起動トリガを終了する)
    [b:GB62234] = 0;                // 運転準備中フラグのクリア
    [b:GB62235] = 1;                // タッチパネル運転中フラグ(1:運転中)
}

// 23 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62243] == 1 && [b:GB62244] == 0){

    [b:GB62244] = 1;                    // 運転準備中フラグ(1:運転準備中)
   bmov([1-23:u32:404449],[u32:GD63402],16);    // NET-IN0～15 のバックアップ
   bmov([1-23:u32:404353],[u32:GD63434],8);     // IN0～7 のバックアップ
    [s32:GD63452] = [1-23:s32:400909];      // プリセット位置のバックアップ
    [u32:GD63454] = [1-23:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

   // NET-IN の入力値をクリアする
    [1-23:w:400126] = 0;

   // タッチパネルから運転する為に信号の割当を変更する
    [1-23:u32:404449] = 48;     // NET-IN0  M0
    [1-23:u32:404451] = 49;     // NET-IN1  M1
    [1-23:u32:404453] = 50;     // NET-IN2  M2
    [1-23:u32:404455] = 51;     // NET-IN3  M3
    [1-23:u32:404457] = 52;     // NET-IN4  M4
    [1-23:u32:404459] = 53;     // NET-IN5  M5
    [1-23:u32:404461] = 4;   // NET-IN6  START
    [1-23:u32:404463] = 3;   // NET-IN7  HOME
    [1-23:u32:404465] = 18;     // NET-IN8  STOP
    [1-23:u32:404467] = 9;   // NET-IN9  MS1
    [1-23:u32:404469] = 10;     // NET-IN10 MS2
    [1-23:u32:404471] = 5;   // NET-IN11 SSTART
    [1-23:u32:404473] = 6;   // NET-IN12 +JOG
    [1-23:u32:404475] = 7;   // NET-IN13 -JOG
    [1-23:u32:404477] = 1;   // NET-IN14 FWD
    [1-23:u32:404479] = 2;   // NET-IN15 RVS

    [1-23:u32:404353] = 32;  // IN0 R0
    [1-23:u32:404355] = 33;  // IN1 R1
    [1-23:u32:404357] = 34;  // IN2 R2
    [1-23:u32:404359] = 35;  // IN3 R3
    [1-23:u32:404361] = 36;  // IN4 R4
    [1-23:u32:404363] = 37;  // IN5 R5
    [1-23:u32:404365] = 18;  // IN6 STOP
    [1-23:u32:404367] = 39;  // IN7 R7
```

```
  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63450] = [1-23:s32:400199];
  [1-23:s32:400909] = [s32:GD63450];

  // Configurationの実行
  [1-23:s32:400397] = 1;

  // Configuration実行待ち時間のタイマー記録(単位は秒)
  [w:GD63456] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63456] > 1 && [b:GB62244] == 1){

  [1-23:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-23:s32:400909] = [s32:GD63452]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-23:s32:400397] = 0;          // Configuration実行のゼロクリア
  [1-23:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62243] = 0;             // 運転準備開始トリガのクリア
                         // (このスクリプトの起動トリガを終了する)
  [b:GB62244] = 0;             // 運転準備中フラグのクリア
  [b:GB62245] = 1;             // タッチパネル運転中フラグ(1:運転中)
}

// 24軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62253] == 1 && [b:GB62254] == 0){

  [b:GB62254] = 1;                // 運転準備中フラグ(1:運転準備中)
  bmov([1-24:u32:404449],[u32:GD63502],16);    // NET-IN0~15のバックアップ
  bmov([1-24:u32:404353],[u32:GD63534],8);     // IN0~7のバックアップ
  [s32:GD63552] = [1-24:s32:400909];      // プリセット位置のバックアップ
  [u32:GD63554] = [1-24:u32:404169];      // JOG移動量のバックアップ(GOTの最小移動量に使用)

  // NET-INの入力値をクリアする
  [1-24:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-24:u32:404449] = 48;    // NET-IN0  M0
  [1-24:u32:404451] = 49;    // NET-IN1  M1
  [1-24:u32:404453] = 50;    // NET-IN2  M2
  [1-24:u32:404455] = 51;    // NET-IN3  M3
  [1-24:u32:404457] = 52;    // NET-IN4  M4
  [1-24:u32:404459] = 53;    // NET-IN5  M5
  [1-24:u32:404461] = 4;   // NET-IN6  START
  [1-24:u32:404463] = 3;   // NET-IN7  HOME
  [1-24:u32:404465] = 18;    // NET-IN8  STOP
  [1-24:u32:404467] = 9;   // NET-IN9  MS1
  [1-24:u32:404469] = 10;    // NET-IN10 MS2
  [1-24:u32:404471] = 5;   // NET-IN11 SSTART
  [1-24:u32:404473] = 6;   // NET-IN12 +JOG
  [1-24:u32:404475] = 7;   // NET-IN13 -JOG
  [1-24:u32:404477] = 1;   // NET-IN14 FWD
  [1-24:u32:404479] = 2;   // NET-IN15 RVS
```

```
  [1-24:u32:404353] = 32;  // IN0 R0
  [1-24:u32:404355] = 33;  // IN1 R1
  [1-24:u32:404357] = 34;  // IN2 R2
  [1-24:u32:404359] = 35;  // IN3 R3
  [1-24:u32:404361] = 36;  // IN4 R4
  [1-24:u32:404363] = 37;  // IN5 R5
  [1-24:u32:404365] = 18;  // IN6 STOP
  [1-24:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63550] = [1-24:s32:400199];
  [1-24:s32:400909] = [s32:GD63550];

  // Configurationの実行
  [1-24:s32:400397] = 1;

  // Configuration実行待ち時間のタイマー記録(単位は秒)
  [w:GD63556] = [w:GS7];
}

// Configuration実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63556] > 1 && [b:GB62254] == 1){

  [1-24:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-24:s32:400909] = [s32:GD63552]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-24:s32:400397] = 0;          // Configuration実行のゼロクリア
  [1-24:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62253] = 0;              // 運転準備開始トリガのクリア
                          // (このスクリプトの起動トリガを終了する)
  [b:GB62254] = 0;              // 運転準備中フラグのクリア
  [b:GB62255] = 1;              // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプトNo.	31038	スクリプト名	Script31038
コメント	17-20_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62225

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データNo.をM0～M5(割付直したNET-IN)に反映させる
// 反映したM0～M5は、ドライバ入力指令(400126)にて、ONする

// 21-24_マルチ運転中スクリプト No.31038
//21-24_マルチ運転準備スクリプト(No.31037)のGB62225、GB62235、GB62245、GB62255のON中で、スクリプトを起動

// [b:GB62225], [b:GB62235], [b:GB62245], [b:GB62255] : タッチパネル運転中フラグ(このスクリプトの起動トリガ)
//また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-21:u32:400195] ～ [1-24:u32:400195] : モニタの現在の選択データNo.のModbusアドレス
// [u32:GD63200], [u32GD63300], [GD63400], [u32:GD63500] : 運転データNo.のデバイス

// 選択した運転データNo.をドライバ入力指令のM0～M5に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データNo.(400195)と選択した運転データNo.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる
```

```
// 21 軸目
if([b:GB62225] == 1){

   if([1-21:u32:400195] != [u32:GD63200]){
      [1-21:b:400126.b0] = [b:GD63200.b0];    // M0
      [1-21:b:400126.b1] = [b:GD63200.b1];    // M1
      [1-21:b:400126.b2] = [b:GD63200.b2];    // M2
      [1-21:b:400126.b3] = [b:GD63200.b3];    // M3
      [1-21:b:400126.b4] = [b:GD63200.b4];    // M4
      [1-21:b:400126.b5] = [b:GD63200.b5];    // M5
   }
}

// 22 軸目
if([b:GB62235] == 1){

   if([1-22:u32:400195] != [u32:GD63300]){
      [1-22:b:400126.b0] = [b:GD63300.b0];    // M0
      [1-22:b:400126.b1] = [b:GD63300.b1];    // M1
      [1-22:b:400126.b2] = [b:GD63300.b2];    // M2
      [1-22:b:400126.b3] = [b:GD63300.b3];    // M3
      [1-22:b:400126.b4] = [b:GD63300.b4];    // M4
      [1-22:b:400126.b5] = [b:GD63300.b5];    // M5
   }
}

// 23 軸目
if([b:GB62245] == 1){

   if([1-23:u32:400195] != [u32:GD63400]){
      [1-23:b:400126.b0] = [b:GD63400.b0];    // M0
      [1-23:b:400126.b1] = [b:GD63400.b1];    // M1
      [1-23:b:400126.b2] = [b:GD63400.b2];    // M2
      [1-23:b:400126.b3] = [b:GD63400.b3];    // M3
      [1-23:b:400126.b4] = [b:GD63400.b4];    // M4
      [1-23:b:400126.b5] = [b:GD63400.b5];    // M5
   }
}

// 24 軸目
if([b:GB62255] == 1){

   if([1-24:u32:400195] != [u32:GD63500]){
      [1-24:b:400126.b0] = [b:GD63500.b0];    // M0
      [1-24:b:400126.b1] = [b:GD63500.b1];    // M1
      [1-24:b:400126.b2] = [b:GD63500.b2];    // M2
      [1-24:b:400126.b3] = [b:GD63500.b3];    // M3
      [1-24:b:400126.b4] = [b:GD63500.b4];    // M4
      [1-24:b:400126.b5] = [b:GD63500.b5];    // M5
   }
}
```

スクリプト No.	31039	スクリプト名	Script31039
コメント	21-24_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62226

```
// タッチパネルからの運転を終了するスクリプト
// バックアップした NET-IN などの信号を元に戻したり、必要なパラメータなどを元に戻す
// 21-24 軸のマルチ運転の画面スクリプト

// 21-24_マルチ運転終了スクリプト No.31039
// 21-24_マルチ運転開始_終了前確認スクリプト(No.31036)の GB62226、GB62236、GB62246、GB62256 の ON
中で、スクリプトを起動

// [b:GB62225],    [b:GB62235],    [b:GB62245],    [b:GB62255]    : タッチパネル運転中フラグ
// [b:GB62226],    [b:GB62236],    [b:GB62246],    [b:GB62256]    : 運転終了開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62227],    [b:GB62237],    [b:GB62247],    [b:GB62257]    : 運転終了中フラグ
// [u32:GD63202], [u32:GD63302], [u32:GD63402], [u32:GD63502] : NET-IN0～15 のバックアップデバイ
ス
// [u32:GD63234], [u32:GD63334], [u32:GD63434], [u32:GD63534] : IN0～7 のバックアップデバイス
// [s32:GD63250], [s32:GD63350], [s32:GD63450], [s32:GD63550] : モニタの指令位置のバックアップ
デバイス
// [s32:GD63252], [s32:GD63352], [s32:GD63452], [s32:GD63552] : プリセット位置のバックアップデ
バイス
// [u32:GD63254], [u32:GD63354], [u32:GD63454], [u32:GD63554] : JOG 移動量のバックアップデバイス
(GOT の最小移動量に使用)
// [w:GD63256],   [w:GD63356],    [w:GD63456],   [w:GD63556]     : Configuration 実行待ち時間用
タイマー

// 21 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62226] == 1 && [b:GB62227] == 0){

  [b:GB62225] = 0;                      // タッチパネル運転中フラグのクリア
  [b:GB62227] = 1;                      // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63202],[1-21:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63234],[1-21:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-21:u32:404169] = [u32:GD63254];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63250] = [1-21:s32:400199];
  [1-21:s32:400909] = [s32:GD63250];

  // NET-IN の入力値をクリアする
  [1-21:w:400126] = 0;

  // Configuration の実行
  [1-21:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63256] > 1 && [b:GB62227] == 1){

  [1-21:s32:400395] = 1;           // P-PRESET(プリセット)実行する
  [1-21:s32:400909] = [s32:GD63252]; // プリセット位置パラメータを元に戻す
  [1-21:s32:400397] = 0;           // Configuration 実行のゼロクリア
  [1-21:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア
```

```
  [b:GB62226] = 0;              // 運転終了開始トリガのクリア
                             // (このスクリプトの起動トリガを終了する)
  [b:GB62227] = 0;              // 運転終了中フラグのクリア
}

// 22 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62236] == 1 && [b:GB62237] == 0){

  [b:GB62235] = 0;                   // タッチパネル運転中フラグのクリア
  [b:GB62237] = 1;                   // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63302],[1-22:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63334],[1-22:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-22:u32:404169] = [u32:GD63354];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63350] = [1-22:s32:400199];
  [1-22:s32:400909] = [s32:GD63350];

  // NET-IN の入力値をクリアする
  [1-22:w:400126] = 0;

  // Configuration の実行
  [1-22:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63356] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63356] > 1 && [b:GB62237] == 1){

  [1-22:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-22:s32:400909] = [s32:GD63352]; // プリセット位置パラメータを元に戻す
  [1-22:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-22:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62236] = 0;              // 運転終了開始トリガのクリア
                             // (このスクリプトの起動トリガを終了する)
  [b:GB62237] = 0;              // 運転終了中フラグのクリア
}

// 23 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62246] == 1 && [b:GB62247] == 0){

  [b:GB62245] = 0;                   // タッチパネル運転中フラグのクリア
  [b:GB62247] = 1;                   // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63402],[1-23:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63434],[1-23:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-23:u32:404169] = [u32:GD63454];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63450] = [1-23:s32:400199];
  [1-23:s32:400909] = [s32:GD63450];
```

```
  // NET-IN の入力値をクリアする
  [1-23:w:400126] = 0;

  // Configuration の実行
  [1-23:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63456] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63456] > 1 && [b:GB62247] == 1){

  [1-23:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-23:s32:400909] = [s32:GD63452]; // プリセット位置パラメータを元に戻す
  [1-23:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-23:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62246] = 0;              // 運転終了開始トリガのクリア
                            // (このスクリプトの起動トリガを終了する)
  [b:GB62247] = 0;              // 運転終了中フラグのクリア
}

// 24 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62256] == 1 && [b:GB62257] == 0){

  [b:GB62255] = 0;                     // タッチパネル運転中フラグのクリア
  [b:GB62257] = 1;                     // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63502],[1-24:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63534],[1-24:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-24:u32:404169] = [u32:GD63554];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63550] = [1-24:s32:400199];
  [1-24:s32:400909] = [s32:GD63550];

  // NET-IN の入力値をクリアする
  [1-24:w:400126] = 0;

  // Configuration の実行
  [1-24:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63556] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63556] > 1 && [b:GB62257] == 1){

  [1-24:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-24:s32:400909] = [s32:GD63552]; // プリセット位置パラメータを元に戻す
  [1-24:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-24:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62256] = 0;              // 運転終了開始トリガのクリア
```

```
                    // (このスクリプトの起動トリガを終了する)
   [b:GB62257] = 0;            // 運転終了中フラグのクリア
}
```

スクリプトNo.	31036	スクリプト名	Script31036
コメント	21-24_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62230

ベース画面31027のスクリプトNo.31036と同じです。

スクリプトNo.	31037	スクリプト名	Script31037
コメント	21-24_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62233

ベース画面31027のスクリプトNo.31037と同じです。

スクリプトNo.	31038	スクリプト名	Script31038
コメント	21-24_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62235

ベース画面31027のスクリプトNo.31038と同じです。

スクリプトNo.	31039	スクリプト名	Script31039
コメント	21-24_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62236

ベース画面31027のスクリプトNo.31039と同じです。

スクリプトNo.	31036	スクリプト名	Script31036
コメント	21-24_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62240

ベース画面31027のスクリプトNo.31036と同じです。

スクリプトNo.	31037	スクリプト名	Script31037
コメント	21-24_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62243

ベース画面31027のスクリプトNo.31037と同じです。

スクリプトNo.	31038	スクリプト名	Script31038
コメント	21-24_マルチ運転中		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62245

ベース画面31027のスクリプトNo.31038と同じです。

スクリプトNo.	31039	スクリプト名	Script31039
コメント	21-24_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62246

ベース画面31027のスクリプトNo.31039と同じです。

スクリプトNo.	31036	スクリプト名	Script31036
コメント	21-24_マルチ運転開始_終了前確認		
データ形式	符号付きBIN16	トリガ種別	ON中周期 2(秒) GB62250

ベース画面31027のスクリプトNo.31036と同じです。

スクリプトNo.	31037	スクリプト名	Script31037
コメント	21-24_マルチ運転準備		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62253

ベース画面 31027 のスクリプト No.31037 と同じです。			
スクリプト No.	31038	スクリプト名	Script31038
コメント	21-24_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62255
ベース画面 31027 のスクリプト No.31038 と同じです。			
スクリプト No.	31039	スクリプト名	Script31039
コメント	21-24_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62256
ベース画面 31027 のスクリプト No.31039 と同じです。			

ベース画面 31028

スクリプト No.	31040	スクリプト名	Script31040
コメント	25-28_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62260

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 25-28 軸のマルチ運転の画面スクリプト

// 25-28_マルチ運転開始_終了前確認スクリプト No.31040
// 25-28 軸のマルチ運転画面のそれぞれの開始ボタンの ON 中周期、2 秒で、スクリプトを起動

// [b:GB62260] : 25 軸目の起動トリガ、ビットセットで起動
// [b:GB62270] : 26 軸目の起動トリガ、ビットセットで起動
// [b:GB62280] : 27 軸目の起動トリガ、ビットセットで起動
// [b:GB62290] : 28 軸目の起動トリガ、ビットセットで起動
// [b:GB62265] : 25 軸目のタッチパネル運転中フラグ
// [b:GB62275] : 26 軸目のタッチパネル運転中フラグ
// [b:GB62285] : 27 軸目のタッチパネル運転中フラグ
// [b:GB62295] : 28 軸目のタッチパネル運転中フラグ

// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 25 軸目
if([b:GB62260] == 1){
  [w:GD61000] = 25;

  // アラームチェック
  if([1-25:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62261] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62261] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-25:s32:400201] == 0){   // 運転していない場合
    [b:GB62262] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62262] = 0;        // 運転中確認フラグ(0:運転中)
```

```
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62261] == 1 && [b:GB62262] == 1){
    if([b:GB62265] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62260] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62263] = 1;    // 運転準備開始トリガ
      [b:GB62261] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62262] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62260] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62266] = 1;    // 運転終了開始トリガ
      [b:GB62261] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62262] = 0;    // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62260] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62261] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62262] = 0;    // 運転中確認フラグのクリア
  }
}

// 26 軸目
if([b:GB62270] == 1){
  [w:GD61000] = 26;

  // アラームチェック
  if([1-26:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62271] = 1;       // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;       // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                 // アラームが発生している場合
    [b:GB62271] = 0;       // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;     // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-26:s32:400201] == 0){   // 運転していない場合
    [b:GB62272] = 1;       // 運転中確認フラグ(1:運転していない)
  } else {                 // 運転中の場合
    [b:GB62272] = 0;       // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;     // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62271] == 1 && [b:GB62272] == 1){
    if([b:GB62275] == 0){    // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62270] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62273] = 1;    // 運転準備開始トリガ
      [b:GB62271] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62272] = 0;    // 運転中確認フラグのクリア
    } else {              // タッチパネル運転中なら、運転終了準備をする
      [b:GB62270] = 0;    // このスクリプトの起動トリガを終了する
      [b:GB62276] = 1;    // 運転終了開始トリガ
      [b:GB62271] = 0;    // アラーム発生状態確認フラグのクリア
      [b:GB62272] = 0;    // 運転中確認フラグのクリア
```

```
      }
  } else {
        [b:GB62270] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62271] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62272] = 0;    // 運転中確認フラグのクリア
  }
}

// 27 軸目
if([b:GB62280] == 1){
  [w:GD61000] = 27;

  // アラームチェック
  if([1-27:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62281] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62281] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-27:s32:400201] == 0){   // 運転していない場合
    [b:GB62282] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62282] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62281] == 1 && [b:GB62282] == 1){
    if([b:GB62285] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
      [b:GB62280] = 0;      // このスクリプトの起動トリガを終了する
      [b:GB62283] = 1;      // 運転準備開始トリガ
      [b:GB62281] = 0;      // アラーム発生状態確認フラグのクリア
      [b:GB62282] = 0;      // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
      [b:GB62280] = 0;      // このスクリプトの起動トリガを終了する
      [b:GB62286] = 1;      // 運転終了開始トリガ
      [b:GB62281] = 0;      // アラーム発生状態確認フラグのクリア
      [b:GB62282] = 0;      // 運転中確認フラグのクリア
    }
  } else {
      [b:GB62280] = 0;      // このスクリプトの起動トリガを終了する
      [b:GB62281] = 0;      // アラーム発生状態確認フラグのクリア
      [b:GB62282] = 0;      // 運転中確認フラグのクリア
  }
}

// 28 軸目
if([b:GB62290] == 1){
  [w:GD61000] = 28;

  // アラームチェック
  if([1-28:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62291] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
```

```
      [w:GD60004] = 0;          // アラーム発生中確認オーバーラップウィンドウの消去
   } else {                     // アラームが発生している場合
      [b:GB62291] = 0;          // アラーム発生状態確認フラグ(0:アラーム発生中)
      [w:GD60004] = 32007;        // アラーム発生中確認オーバーラップウィンドウの表示
   }

   // 運転中チェック
   if([1-28:s32:400201] == 0){   // 運転していない場合
      [b:GB62292] = 1;          // 運転中確認フラグ(1:運転していない)
   } else {                     // 運転中の場合
      [b:GB62292] = 0;          // 運転中確認フラグ(0:運転中)
      [w:GD60004] = 32008;        // モーター運転中確認オーバーラップウィンドウの表示
   }

   // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
   if([b:GB62291] == 1 && [b:GB62292] == 1){
      if([b:GB62295] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
         [b:GB62290] = 0;       // このスクリプトの起動トリガを終了する
         [b:GB62293] = 1;       // 運転準備開始トリガ
         [b:GB62291] = 0;       // アラーム発生状態確認フラグのクリア
         [b:GB62292] = 0;       // 運転中確認フラグのクリア
      } else {                  // タッチパネル運転中なら、運転終了準備をする
         [b:GB62290] = 0;       // このスクリプトの起動トリガを終了する
         [b:GB62296] = 1;       // 運転終了開始トリガ
         [b:GB62291] = 0;       // アラーム発生状態確認フラグのクリア
         [b:GB62292] = 0;       // 運転中確認フラグのクリア
      }
   } else {
         [b:GB62290] = 0;       // このスクリプトの起動トリガを終了する
         [b:GB62291] = 0;       // アラーム発生状態確認フラグのクリア
         [b:GB62292] = 0;       // 運転中確認フラグのクリア
   }
}
```

スクリプト No.	31041	スクリプト名	Script31041
コメント	25-28_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62263

```
// タッチパネルから運転できるように準備するスクリプト
// NET-IN などの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 25-28 軸マルチ運転の画面スクリプト

// 25-28_マルチ運転準備スクリプト No.31041
// 25-28_マルチ運転開始_終了前確認スクリプト(No.31040)の GB62263、GB62273、GB62283、GB62293 の ON
中で、スクリプトを起動

// [b:GB62263],    [b:GB62273],    [b:GB62283],    [b:GB62293]   : 運転準備開始トリガ(このスクリ
プトの起動トリガ)
// [b:GB62264],    [b:GB62274],    [b:GB62284],    [b:GB62294]   : 運転準備中フラグ
// [b:GB62265],    [b:GB62275],    [b:GB62285],    [b:GB62295]   : タッチパネル運転中フラグ
// [u32:GD63602], [u32:GD63702], [u32:GD63802], [u32:GD63902] : NET-IN0～15 のバックアップデバイ
ス
// [u32:GD63634], [u32:GD63734], [u32:GD63834], [u32:GD63934]  : IN0～7 のバックアップデバイス
// [s32:GD63650], [s32:GD63750], [s32:GD63850], [s32:GD63950]  : モニタの指令位置のバックアップ
デバイス
// [s32:GD63652], [s32:GD63752], [s32:GD63852], [u32:GD63952]  : プリセット位置のバックアップデ
バイス
// [u32:GD63654], [u32:GD63754], [u32:GD63854], [u32:GD63954] : JOG 移動量のバックアップデバイス
```

```
(GOT の最小移動量に使用)
// [w:GD63656], [w:GD63756], [w:GD63856], [w:GD63956]   : Configuration 実行待ち時間用
タイマー

// 25 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62263] == 1 && [b:GB62264] == 0){

  [b:GB62264] = 1;                     // 運転準備中フラグ(1:運転準備中)
  bmov([1-25:u32:404449],[u32:GD63602],16);    // NET-IN0～15 のバックアップ
  bmov([1-25:u32:404353],[u32:GD63634],8);     // IN0～7 のバックアップ
  [s32:GD63652] = [1-25:s32:400909];      // プリセット位置のバックアップ
  [u32:GD63654] = [1-25:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-25:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-25:u32:404449] = 48;    // NET-IN0  M0
  [1-25:u32:404451] = 49;    // NET-IN1  M1
  [1-25:u32:404453] = 50;    // NET-IN2  M2
  [1-25:u32:404455] = 51;    // NET-IN3  M3
  [1-25:u32:404457] = 52;    // NET-IN4  M4
  [1-25:u32:404459] = 53;    // NET-IN5  M5
  [1-25:u32:404461] = 4;  // NET-IN6  START
  [1-25:u32:404463] = 3;  // NET-IN7  HOME
  [1-25:u32:404465] = 18;    // NET-IN8  STOP
  [1-25:u32:404467] = 9;  // NET-IN9  MS1
  [1-25:u32:404469] = 10;    // NET-IN10 MS2
  [1-25:u32:404471] = 5;  // NET-IN11 SSTART
  [1-25:u32:404473] = 6;  // NET-IN12 +JOG
  [1-25:u32:404475] = 7;  // NET-IN13 -JOG
  [1-25:u32:404477] = 1;  // NET-IN14 FWD
  [1-25:u32:404479] = 2;  // NET-IN15 RVS

  [1-25:u32:404353] = 32;  // IN0 R0
  [1-25:u32:404355] = 33;  // IN1 R1
  [1-25:u32:404357] = 34;  // IN2 R2
  [1-25:u32:404359] = 35;  // IN3 R3
  [1-25:u32:404361] = 36;  // IN4 R4
  [1-25:u32:404363] = 37;  // IN5 R5
  [1-25:u32:404365] = 18;  // IN6 STOP
  [1-25:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63650] = [1-25:s32:400199];
  [1-25:s32:400909] = [s32:GD63650];

  // Configuration の実行
  [1-25:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63656] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
```

```
if([w:GS7] - [w:GD63656] > 1 && [b:GB62264] == 1){

  [1-25:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-25:s32:400909] = [s32:GD63652]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-25:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-25:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62263] = 0;              // 運転準備開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62264] = 0;              // 運転準備中フラグのクリア
  [b:GB62265] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 26 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62273] == 1 && [b:GB62274] == 0){

  [b:GB62274] = 1;                    // 運転準備中フラグ(1:運転準備中)
  bmov([1-26:u32:404449],[u32:GD63702],16);    // NET-IN0～15 のバックアップ
  bmov([1-26:u32:404353],[u32:GD63734],8);     // IN0～7 のバックアップ
  [s32:GD63752] = [1-26:s32:400909];       // プリセット位置のバックアップ
  [u32:GD63754] = [1-26:u32:404169];       // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-26:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-26:u32:404449] = 48;     // NET-IN0  M0
  [1-26:u32:404451] = 49;     // NET-IN1  M1
  [1-26:u32:404453] = 50;     // NET-IN2  M2
  [1-26:u32:404455] = 51;     // NET-IN3  M3
  [1-26:u32:404457] = 52;     // NET-IN4  M4
  [1-26:u32:404459] = 53;     // NET-IN5  M5
  [1-26:u32:404461] = 4;   // NET-IN6  START
  [1-26:u32:404463] = 3;   // NET-IN7  HOME
  [1-26:u32:404465] = 18;     // NET-IN8  STOP
  [1-26:u32:404467] = 9;   // NET-IN9  MS1
  [1-26:u32:404469] = 10;     // NET-IN10 MS2
  [1-26:u32:404471] = 5;   // NET-IN11 SSTART
  [1-26:u32:404473] = 6;   // NET-IN12 +JOG
  [1-26:u32:404475] = 7;   // NET-IN13 -JOG
  [1-26:u32:404477] = 1;   // NET-IN14 FWD
  [1-26:u32:404479] = 2;   // NET-IN15 RVS

  [1-26:u32:404353] = 32;  // IN0 R0
  [1-26:u32:404355] = 33;  // IN1 R1
  [1-26:u32:404357] = 34;  // IN2 R2
  [1-26:u32:404359] = 35;  // IN3 R3
  [1-26:u32:404361] = 36;  // IN4 R4
  [1-26:u32:404363] = 37;  // IN5 R5
  [1-26:u32:404365] = 18;  // IN6 STOP
  [1-26:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63750] = [1-26:s32:400199];
  [1-26:s32:400909] = [s32:GD63750];
```

```
  // Configuration の実行
  [1-26:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63756] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63756] > 1 && [b:GB62274] == 1){

  [1-26:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-26:s32:400909] = [s32:GD63752]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-26:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-26:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62273] = 0;              // 運転準備開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62274] = 0;              // 運転準備中フラグのクリア
  [b:GB62275] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 27 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62283] == 1 && [b:GB62284] == 0){

  [b:GB62284] = 1;                    // 運転準備中フラグ(1:運転準備中)
  bmov([1-27:u32:404449],[u32:GD63802],16);    // NET-IN0～15 のバックアップ
  bmov([1-27:u32:404353],[u32:GD63834],8);     // IN0～7 のバックアップ
  [s32:GD63852] = [1-27:s32:400909];      // プリセット位置のバックアップ
  [u32:GD63854] = [1-27:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-27:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-27:u32:404449] = 48;    // NET-IN0  M0
  [1-27:u32:404451] = 49;    // NET-IN1  M1
  [1-27:u32:404453] = 50;    // NET-IN2  M2
  [1-27:u32:404455] = 51;    // NET-IN3  M3
  [1-27:u32:404457] = 52;    // NET-IN4  M4
  [1-27:u32:404459] = 53;    // NET-IN5  M5
  [1-27:u32:404461] = 4;   // NET-IN6  START
  [1-27:u32:404463] = 3;   // NET-IN7  HOME
  [1-27:u32:404465] = 18;    // NET-IN8  STOP
  [1-27:u32:404467] = 9;   // NET-IN9  MS1
  [1-27:u32:404469] = 10;    // NET-IN10 MS2
  [1-27:u32:404471] = 5;   // NET-IN11 SSTART
  [1-27:u32:404473] = 6;   // NET-IN12 +JOG
  [1-27:u32:404475] = 7;   // NET-IN13 -JOG
  [1-27:u32:404477] = 1;   // NET-IN14 FWD
  [1-27:u32:404479] = 2;   // NET-IN15 RVS

  [1-27:u32:404353] = 32;  // IN0 R0
  [1-27:u32:404355] = 33;  // IN1 R1
  [1-27:u32:404357] = 34;  // IN2 R2
```

```
  [1-27:u32:404359] = 35;  // IN3 R3
  [1-27:u32:404361] = 36;  // IN4 R4
  [1-27:u32:404363] = 37;  // IN5 R5
  [1-27:u32:404365] = 18;  // IN6 STOP
  [1-27:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63850] = [1-27:s32:400199];
  [1-27:s32:400909] = [s32:GD63850];

  // Configuration の実行
  [1-27:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63856] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63856] > 1 && [b:GB62284] == 1){

  [1-27:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-27:s32:400909] = [s32:GD63852]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-27:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-27:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62283] = 0;              // 運転準備開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62284] = 0;              // 運転準備中フラグのクリア
  [b:GB62285] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 28 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62293] == 1 && [b:GB62294] == 0){

  [b:GB62294] = 1;                   // 運転準備中フラグ(1:運転準備中)
  bmov([1-28:u32:404449],[u32:GD63902],16);    // NET-IN0～15 のバックアップ
  bmov([1-28:u32:404353],[u32:GD63934],8);     // IN0～7 のバックアップ
  [s32:GD63952] = [1-28:s32:400909];      // プリセット位置のバックアップ
  [u32:GD63954] = [1-28:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-28:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-28:u32:404449] = 48;    // NET-IN0  M0
  [1-28:u32:404451] = 49;    // NET-IN1  M1
  [1-28:u32:404453] = 50;    // NET-IN2  M2
  [1-28:u32:404455] = 51;    // NET-IN3  M3
  [1-28:u32:404457] = 52;    // NET-IN4  M4
  [1-28:u32:404459] = 53;    // NET-IN5  M5
  [1-28:u32:404461] = 4;   // NET-IN6  START
  [1-28:u32:404463] = 3;   // NET-IN7  HOME
  [1-28:u32:404465] = 18;    // NET-IN8  STOP
  [1-28:u32:404467] = 9;   // NET-IN9  MS1
  [1-28:u32:404469] = 10;    // NET-IN10 MS2
```

```
   [1-28:u32:404471] = 5;   // NET-IN11 SSTART
   [1-28:u32:404473] = 6;   // NET-IN12 +JOG
   [1-28:u32:404475] = 7;   // NET-IN13 -JOG
   [1-28:u32:404477] = 1;   // NET-IN14 FWD
   [1-28:u32:404479] = 2;   // NET-IN15 RVS

   [1-28:u32:404353] = 32;  // IN0 R0
   [1-28:u32:404355] = 33;  // IN1 R1
   [1-28:u32:404357] = 34;  // IN2 R2
   [1-28:u32:404359] = 35;  // IN3 R3
   [1-28:u32:404361] = 36;  // IN4 R4
   [1-28:u32:404363] = 37;  // IN5 R5
   [1-28:u32:404365] = 18;  // IN6 STOP
   [1-28:u32:404367] = 39;  // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD63950] = [1-28:s32:400199];
   [1-28:s32:400909] = [s32:GD63950];

   // Configuration の実行
   [1-28:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD63956] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD63956] > 1 && [b:GB62294] == 1){

   [1-28:s32:400395] = 1;          // P-PRESET(プリセット)実行する
   [1-28:s32:400909] = [s32:GD63952]; // バックアップしたプリセット位置パラメータを元に戻す
   [1-28:s32:400397] = 0;          // Configuration 実行のゼロクリア
   [1-28:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62293] = 0;              // 運転準備開始トリガのクリア
                            // (このスクリプトの起動トリガを終了する)
   [b:GB62294] = 0;              // 運転準備中フラグのクリア
   [b:GB62295] = 1;              // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプト No.	31042	スクリプト名	Script31042
コメント	25-28_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62265

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データ No.を M0～M5(割付直した NET-IN)に反映させる
// 反映した M0～M5 は、ドライバ入力指令(400126)にて、ON する

// 25-28_マルチ運転中スクリプト No.31042
// 25-28_マルチ運転準備スクリプト(No.31041)の GB62265、GB62275、GB62285、GB62295 の ON 中で、スクリプトを起動

// [b:GB62265], [b:GB62275], [b:GB62285], [b:GB62295] : タッチパネル運転中フラグ(このスクリプトの起動トリガ)
// また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-25:u32:400195] ～ [1-28:u32:400195] : モニタの現在の選択データ No.の Modbus アドレス
// [u32:GD63600], [u32GD63700], [GD63800], [u32:GD63900] : 運転データ No.のデバイス
```

```
// 選択した運転データNo.をドライバ入力指令のM0～M5に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データNo.(400195)と選択した運転データNo.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

// 25軸目
if([b:GB62265] == 1){

   if([1-25:u32:400195] != [u32:GD63600]){
      [1-25:b:400126.b0] = [b:GD63600.b0];    // M0
      [1-25:b:400126.b1] = [b:GD63600.b1];    // M1
      [1-25:b:400126.b2] = [b:GD63600.b2];    // M2
      [1-25:b:400126.b3] = [b:GD63600.b3];    // M3
      [1-25:b:400126.b4] = [b:GD63600.b4];    // M4
      [1-25:b:400126.b5] = [b:GD63600.b5];    // M5
   }
}

// 26軸目
if([b:GB62275] == 1){

   if([1-26:u32:400195] != [u32:GD63700]){
      [1-26:b:400126.b0] = [b:GD63700.b0];    // M0
      [1-26:b:400126.b1] = [b:GD63700.b1];    // M1
      [1-26:b:400126.b2] = [b:GD63700.b2];    // M2
      [1-26:b:400126.b3] = [b:GD63700.b3];    // M3
      [1-26:b:400126.b4] = [b:GD63700.b4];    // M4
      [1-26:b:400126.b5] = [b:GD63700.b5];    // M5
   }
}

// 27軸目
if([b:GB62285] == 1){

   if([1-27:u32:400195] != [u32:GD63800]){
      [1-27:b:400126.b0] = [b:GD63800.b0];    // M0
      [1-27:b:400126.b1] = [b:GD63800.b1];    // M1
      [1-27:b:400126.b2] = [b:GD63800.b2];    // M2
      [1-27:b:400126.b3] = [b:GD63800.b3];    // M3
      [1-27:b:400126.b4] = [b:GD63800.b4];    // M4
      [1-27:b:400126.b5] = [b:GD63800.b5];    // M5
   }
}

// 28軸目
if([b:GB62295] == 1){

   if([1-28:u32:400195] != [u32:GD63900]){
      [1-28:b:400126.b0] = [b:GD63900.b0];    // M0
      [1-28:b:400126.b1] = [b:GD63900.b1];    // M1
      [1-28:b:400126.b2] = [b:GD63900.b2];    // M2
      [1-28:b:400126.b3] = [b:GD63900.b3];    // M3
      [1-28:b:400126.b4] = [b:GD63900.b4];    // M4
      [1-28:b:400126.b5] = [b:GD63900.b5];    // M5
   }
```

```
}
```

スクリプト No.	31043	スクリプト名	Script31043
コメント	25-28_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62266

```
// タッチパネルからの運転を終了するスクリプト
// バックアップした NET-IN などの信号を元に戻したり、必要なパラメータなどを元に戻す
// 25-28 軸のマルチ運転の画面スクリプト

// 25-28_マルチ運転終了スクリプト No.31043
// 25-28_マルチ運転開始_終了前確認スクリプト(No.31040)の GB62266、GB62276、GB62286、GB62296 の ON 中で、スクリプトを起動

// [b:GB62265],    [b:GB62275],    [b:GB62285],    [b:GB62295]    : タッチパネル運転中フラグ
// [b:GB62266],    [b:GB62276],    [b:GB62286],    [b:GB62296]    : 運転終了開始トリガ(このスクリプトの起動トリガ)
// [b:GB62267],    [b:GB62277],    [b:GB62287],    [b:GB62297]    : 運転終了中フラグ
// [u32:GD63602], [u32:GD63702], [u32:GD63802], [u32:GD63902] : NET-IN0～15 のバックアップデバイス
// [u32:GD63634], [u32:GD63734], [u32:GD63834], [u32:GD63934] : IN0～7 のバックアップデバイス
// [s32:GD63650], [s32:GD63750], [s32:GD63850], [s32:GD63950] : モニタの指令位置のバックアップデバイス
// [s32:GD63652], [s32:GD63752], [s32:GD63852], [s32:GD63952] : プリセット位置のバックアップデバイス
// [u32:GD63654], [u32:GD63754], [u32:GD63854], [u32:GD63954] : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD63656],   [w:GD63756],    [w:GD63856],   [w:GD63956]     : Configuration 実行待ち時間用タイマー

// 25 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62266] == 1 && [b:GB62267] == 0){

  [b:GB62265] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62267] = 1;                    // 運転終了中フラグ(1：運転終了中)
  bmov([u32:GD63602],[1-25:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63634],[1-25:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-25:u32:404169] = [u32:GD63654];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63650] = [1-25:s32:400199];
  [1-25:s32:400909] = [s32:GD63650];

  // NET-IN の入力値をクリアする
  [1-25:w:400126] = 0;

  // Configuration の実行
  [1-25:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63656] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63656] > 1 && [b:GB62267] == 1){
```

```
  [1-25:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-25:s32:400909] = [s32:GD63652]; // プリセット位置パラメータを元に戻す
  [1-25:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-25:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62266] = 0;              // 運転終了開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62267] = 0;              // 運転終了中フラグのクリア
}

// 26 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62276] == 1 && [b:GB62277] == 0){

  [b:GB62275] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62277] = 1;                    // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63702],[1-26:u32:404449],16);    // バックアップした NET-IN0~15 を元に戻す
  bmov([u32:GD63734],[1-26:u32:404353],8);     // バックアップした IN0~7 を元に戻す
  [1-26:u32:404169] = [u32:GD63754];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63750] = [1-26:s32:400199];
  [1-26:s32:400909] = [s32:GD63750];

  // NET-IN の入力値をクリアする
  [1-26:w:400126] = 0;

  // Configuration の実行
  [1-26:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63756] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63756] > 1 && [b:GB62277] == 1){

  [1-26:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-26:s32:400909] = [s32:GD63752]; // プリセット位置パラメータを元に戻す
  [1-26:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-26:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62276] = 0;              // 運転終了開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62277] = 0;              // 運転終了中フラグのクリア
}

// 27 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62286] == 1 && [b:GB62287] == 0){

  [b:GB62285] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62287] = 1;                    // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63802],[1-27:u32:404449],16);    // バックアップした NET-IN0~15 を元に戻す
  bmov([u32:GD63834],[1-27:u32:404353],8);     // バックアップした IN0~7 を元に戻す
  [1-27:u32:404169] = [u32:GD63854];      // バックアップした JOG 移動量を元に戻す
```

```
  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63850] = [1-27:s32:400199];
  [1-27:s32:400909] = [s32:GD63850];

  // NET-IN の入力値をクリアする
  [1-27:w:400126] = 0;

  // Configuration の実行
  [1-27:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63856] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63856] > 1 && [b:GB62287] == 1){

  [1-27:s32:400395] = 1;           // P-PRESET(プリセット)実行する
  [1-27:s32:400909] = [s32:GD63852]; // プリセット位置パラメータを元に戻す
  [1-27:s32:400397] = 0;           // Configuration 実行のゼロクリア
  [1-27:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62286] = 0;              // 運転終了開始トリガのクリア
                            // (このスクリプトの起動トリガを終了する)
  [b:GB62287] = 0;              // 運転終了中フラグのクリア
}

// 28 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62296] == 1 && [b:GB62297] == 0){

  [b:GB62295] = 0;                  // タッチパネル運転中フラグのクリア
  [b:GB62297] = 1;                  // 運転終了中フラグ(1:運転終了中)
  bmov([u32:GD63902],[1-28:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD63934],[1-28:u32:404353],8);     // バックアップした IN0～7 を元に戻す
  [1-28:u32:404169] = [u32:GD63954];       // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD63950] = [1-28:s32:400199];
  [1-28:s32:400909] = [s32:GD63950];

  // NET-IN の入力値をクリアする
  [1-28:w:400126] = 0;

  // Configuration の実行
  [1-28:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD63956] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD63956] > 1 && [b:GB62297] == 1){

  [1-28:s32:400395] = 1;           // P-PRESET(プリセット)実行する
```

```
  [1-28:s32:400909] = [s32:GD63952]; // プリセット位置パラメータを元に戻す
  [1-28:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-28:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62296] = 0;              // 運転終了開始トリガのクリア
                            //(このスクリプトの起動トリガを終了する)
  [b:GB62297] = 0;              // 運転終了中フラグのクリア
}
```

スクリプト No.	31040	スクリプト名	Script31040
コメント	25-28_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62270

ベース画面 31028 のスクリプト No.31040 と同じです。

スクリプト No.	31041	スクリプト名	Script31041
コメント	25-28_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62273

ベース画面 31028 のスクリプト No.31041 と同じです。

スクリプト No.	31042	スクリプト名	Script31042
コメント	25-28_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62275

ベース画面 31028 のスクリプト No.31042 と同じです。

スクリプト No.	31043	スクリプト名	Script31043
コメント	25-28_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62276

ベース画面 31028 のスクリプト No.31043 と同じです。

スクリプト No.	31040	スクリプト名	Script31040
コメント	25-28_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62280

ベース画面 31028 のスクリプト No.31040 と同じです。

スクリプト No.	31041	スクリプト名	Script31041
コメント	25-28_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62283

ベース画面 31028 のスクリプト No.31041 と同じです。

スクリプト No.	31042	スクリプト名	Script31042
コメント	25-28_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62285

ベース画面 31028 のスクリプト No.31042 と同じです。

スクリプト No.	31043	スクリプト名	Script31043
コメント	25-28_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62286

ベース画面 31028 のスクリプト No.31043 と同じです。

スクリプト No.	31040	スクリプト名	Script31040
コメント	25-28_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62290

ベース画面 31028 のスクリプト No.31040 と同じです。

スクリプト No.	31041	スクリプト名	Script31041
コメント	25-28_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62293

ベース画面 31028 のスクリプト No.31041 と同じです。

スクリプト No.	31042	スクリプト名	Script31042
コメント	25-28_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62295

ベース画面 31028 のスクリプト No.31042 と同じです。

スクリプト No.	31043	スクリプト名	Script31043
コメント	25-28_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62296

ベース画面 31028 のスクリプト No.31043 と同じです。

ベース画面 31029

スクリプト No.	31044	スクリプト名	Script31044
コメント	29-31_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62300

```
// マルチ運転開始・終了前に、アラーム発生や運転中でないことを確認するスクリプト
// 29-31 軸のマルチ運転の画面スクリプト

// 29-31_マルチ運転開始_終了前確認スクリプト No.31044
// 29-31 軸のマルチ運転画面のそれぞれの開始ボタンの ON 中周期、2 秒で、スクリプトを起動

// [b:GB62300] : 29 軸目の起動トリガ、ビットセットで起動
// [b:GB62310] : 30 軸目の起動トリガ、ビットセットで起動
// [b:GB62320] : 31 軸目の起動トリガ、ビットセットで起動
// [b:GB62305] : 29 軸目のタッチパネル運転中フラグ
// [b:GB62315] : 30 軸目のタッチパネル運転中フラグ
// [b:GB62325] : 31 軸目のタッチパネル運転中フラグ

// [w:GD61000] = 1 の意味は、以下の通り
// アラーム発生中確認画面は、局番切り換えを行っているため、
// アラームクリア用に局番(ドライバの号機番号)が必要になるため(従って、軸ごとに設定が必要)
// 局番の戻しは、各画面切り換えデバイスで行っている(GD61199 で戻している)

// 29 軸目
if([b:GB62300] == 1){
  [w:GD61000] = 29;

  // アラームチェック
  if([1-29:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62301] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62301] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
```

```
    if([1-29:s32:400201] == 0){   // 運転していない場合
      [b:GB62302] = 1;       // 運転中確認フラグ(1:運転していない)
    } else {                 // 運転中の場合
      [b:GB62302] = 0;       // 運転中確認フラグ(0:運転中)
      [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
    }

    // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
    if([b:GB62301] == 1 && [b:GB62302] == 1){
      if([b:GB62305] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
        [b:GB62300] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62303] = 1;    // 運転準備開始トリガ
        [b:GB62301] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62302] = 0;    // 運転中確認フラグのクリア
      } else {               // タッチパネル運転中なら、運転終了準備をする
        [b:GB62300] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62306] = 1;    // 運転終了開始トリガ
        [b:GB62301] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62302] = 0;    // 運転中確認フラグのクリア
      }
    } else {
        [b:GB62300] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62301] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62302] = 0;    // 運転中確認フラグのクリア
    }
}

// 30 軸目
if([b:GB62310] == 1){
    [w:GD61000] = 30;

    // アラームチェック
    if([1-30:s32:400129] == 0){   // アラームが発生していない場合
      [b:GB62311] = 1;       // アラーム発生状態確認フラグ(1:アラーム発生なし)
      [w:GD60004] = 0;       // アラーム発生中確認オーバーラップウィンドウの消去
    } else {                 // アラームが発生している場合
      [b:GB62311] = 0;       // アラーム発生状態確認フラグ(0:アラーム発生中)
      [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
    }

    // 運転中チェック
    if([1-30:s32:400201] == 0){   // 運転していない場合
      [b:GB62312] = 1;       // 運転中確認フラグ(1:運転していない)
    } else {                 // 運転中の場合
      [b:GB62312] = 0;       // 運転中確認フラグ(0:運転中)
      [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
    }

    // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
    if([b:GB62311] == 1 && [b:GB62312] == 1){
      if([b:GB62315] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
        [b:GB62310] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62313] = 1;    // 運転準備開始トリガ
        [b:GB62311] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62312] = 0;    // 運転中確認フラグのクリア
      } else {               // タッチパネル運転中なら、運転終了準備をする
```

```
        [b:GB62310] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62316] = 1;    // 運転終了開始トリガ
        [b:GB62311] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62312] = 0;    // 運転中確認フラグのクリア
      }
  } else {
        [b:GB62310] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62311] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62312] = 0;    // 運転中確認フラグのクリア
  }
}

// 31軸目
if([b:GB62320] == 1){
  [w:GD61000] = 31;

  // アラームチェック
  if([1-31:s32:400129] == 0){   // アラームが発生していない場合
    [b:GB62321] = 1;        // アラーム発生状態確認フラグ(1:アラーム発生なし)
    [w:GD60004] = 0;        // アラーム発生中確認オーバーラップウィンドウの消去
  } else {                  // アラームが発生している場合
    [b:GB62321] = 0;        // アラーム発生状態確認フラグ(0:アラーム発生中)
    [w:GD60004] = 32007;      // アラーム発生中確認オーバーラップウィンドウの表示
  }

  // 運転中チェック
  if([1-31:s32:400201] == 0){   // 運転していない場合
    [b:GB62322] = 1;        // 運転中確認フラグ(1:運転していない)
  } else {                  // 運転中の場合
    [b:GB62322] = 0;        // 運転中確認フラグ(0:運転中)
    [w:GD60004] = 32008;      // モーター運転中確認オーバーラップウィンドウの表示
  }

  // 上記のチェック終了後、運転開始なのか運転終了なのかを判別する
  if([b:GB62321] == 1 && [b:GB62322] == 1){
    if([b:GB62325] == 0){     // タッチパネル運転中でないなら、運転開始準備をする
        [b:GB62320] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62323] = 1;    // 運転準備開始トリガ
        [b:GB62321] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62322] = 0;    // 運転中確認フラグのクリア
    } else {                // タッチパネル運転中なら、運転終了準備をする
        [b:GB62320] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62326] = 1;    // 運転終了開始トリガ
        [b:GB62321] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62322] = 0;    // 運転中確認フラグのクリア
    }
  } else {
        [b:GB62320] = 0;    // このスクリプトの起動トリガを終了する
        [b:GB62321] = 0;    // アラーム発生状態確認フラグのクリア
        [b:GB62322] = 0;    // 運転中確認フラグのクリア
  }
}
```

スクリプト No.	31045	スクリプト名	Script31045
コメント	29-31_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62303

```
// タッチパネルから運転できるように準備するスクリプト
// NET-IN などの必要な信号の割付を変更したり、必要なパラメータなどをバックアップする
// 29-31 軸マルチ運転の画面スクリプト

// 29-31_マルチ運転準備スクリプト No.31045
// 29-31_マルチ運転開始_終了前確認スクリプト(No.31044)の GB62303、GB62313、GB62323 の ON 中で、スクリプトを起動

// [b:GB62303],    [b:GB62313],    [b:GB62323]    : 運転準備開始トリガ(このスクリプトの起動トリガ)
// [b:GB62304],    [b:GB62314],    [b:GB62324]    : 運転準備中フラグ
// [b:GB62305],    [b:GB62315],    [b:GB62325]    : タッチパネル運転中フラグ
// [u32:GD64002], [u32:GD64102], [u32:GD64202] : NET-IN0～15 のバックアップデバイス
// [u32:GD64034], [u32:GD64134], [u32:GD64234] : IN0～7 のバックアップデバイス
// [s32:GD64050], [s32:GD64150], [s32:GD64250] : モニタの指令位置のバックアップデバイス
// [s32:GD64052], [s32:GD64152], [s32:GD64252] : プリセット位置のバックアップデバイス
// [u32:GD64054], [u32:GD64154], [u32:GD64254]  : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD64056],   [w:GD64156],    [w:GD64256]    : Configuration 実行待ち時間用タイマー

// 29 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62303] == 1 && [b:GB62304] == 0){

  [b:GB62304] = 1;                    // 運転準備中フラグ(1:運転準備中)
  bmov([1-29:u32:404449],[u32:GD64002],16);    // NET-IN0～15 のバックアップ
  bmov([1-29:u32:404353],[u32:GD64034],8);     // IN0～7 のバックアップ
  [s32:GD64052] = [1-29:s32:400909];      // プリセット位置のバックアップ
  [u32:GD64054] = [1-29:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-29:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-29:u32:404449] = 48;    // NET-IN0  M0
  [1-29:u32:404451] = 49;    // NET-IN1  M1
  [1-29:u32:404453] = 50;    // NET-IN2  M2
  [1-29:u32:404455] = 51;    // NET-IN3  M3
  [1-29:u32:404457] = 52;    // NET-IN4  M4
  [1-29:u32:404459] = 53;    // NET-IN5  M5
  [1-29:u32:404461] = 4;  // NET-IN6  START
  [1-29:u32:404463] = 3;  // NET-IN7  HOME
  [1-29:u32:404465] = 18;    // NET-IN8  STOP
  [1-29:u32:404467] = 9;  // NET-IN9  MS1
  [1-29:u32:404469] = 10;    // NET-IN10 MS2
  [1-29:u32:404471] = 5;  // NET-IN11 SSTART
  [1-29:u32:404473] = 6;  // NET-IN12 +JOG
  [1-29:u32:404475] = 7;  // NET-IN13 -JOG
  [1-29:u32:404477] = 1;  // NET-IN14 FWD
  [1-29:u32:404479] = 2;  // NET-IN15 RVS

  [1-29:u32:404353] = 32; // IN0 R0
  [1-29:u32:404355] = 33; // IN1 R1
  [1-29:u32:404357] = 34; // IN2 R2
  [1-29:u32:404359] = 35; // IN3 R3
  [1-29:u32:404361] = 36; // IN4 R4
```

```
    [1-29:u32:404363] = 37;  // IN5 R5
    [1-29:u32:404365] = 18;  // IN6 STOP
    [1-29:u32:404367] = 39;  // IN7 R7

    // プリセット位置パラメータにモニタの指令位置をセットする
    [s32:GD64050] = [1-29:s32:400199];
    [1-29:s32:400909] = [s32:GD64050];

    // Configuration の実行
    [1-29:s32:400397] = 1;

    // Configuration 実行待ち時間のタイマー記録(単位は秒)
    [w:GD64056] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD64056] > 1 && [b:GB62304] == 1){

    [1-29:s32:400395] = 1;          // P-PRESET(プリセット)実行する
    [1-29:s32:400909] = [s32:GD64052]; // バックアップしたプリセット位置パラメータを元に戻す
    [1-29:s32:400397] = 0;          // Configuration 実行のゼロクリア
    [1-29:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

    [b:GB62303] = 0;             // 運転準備開始トリガのクリア
                                // (このスクリプトの起動トリガを終了する)
    [b:GB62304] = 0;             // 運転準備中フラグのクリア
    [b:GB62305] = 1;             // タッチパネル運転中フラグ(1:運転中)
}

// 30 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62313] == 1 && [b:GB62314] == 0){

    [b:GB62314] = 1;                    // 運転準備中フラグ(1:運転準備中)
    bmov([1-30:u32:404449],[u32:GD64102],16);    // NET-IN0～15 のバックアップ
    bmov([1-30:u32:404353],[u32:GD64134],8);     // IN0～7 のバックアップ
    [s32:GD64152] = [1-30:s32:400909];      // プリセット位置のバックアップ
    [u32:GD64154] = [1-30:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

    // NET-IN の入力値をクリアする
    [1-30:w:400126] = 0;

    // タッチパネルから運転する為に信号の割当を変更する
    [1-30:u32:404449] = 48;     // NET-IN0  M0
    [1-30:u32:404451] = 49;     // NET-IN1  M1
    [1-30:u32:404453] = 50;     // NET-IN2  M2
    [1-30:u32:404455] = 51;     // NET-IN3  M3
    [1-30:u32:404457] = 52;     // NET-IN4  M4
    [1-30:u32:404459] = 53;     // NET-IN5  M5
    [1-30:u32:404461] = 4;   // NET-IN6  START
    [1-30:u32:404463] = 3;   // NET-IN7  HOME
    [1-30:u32:404465] = 18;     // NET-IN8  STOP
    [1-30:u32:404467] = 9;   // NET-IN9  MS1
    [1-30:u32:404469] = 10;     // NET-IN10 MS2
    [1-30:u32:404471] = 5;   // NET-IN11 SSTART
    [1-30:u32:404473] = 6;   // NET-IN12 +JOG
```

```
  [1-30:u32:404475] = 7;   // NET-IN13 -JOG
  [1-30:u32:404477] = 1;   // NET-IN14 FWD
  [1-30:u32:404479] = 2;   // NET-IN15 RVS

  [1-30:u32:404353] = 32;  // IN0 R0
  [1-30:u32:404355] = 33;  // IN1 R1
  [1-30:u32:404357] = 34;  // IN2 R2
  [1-30:u32:404359] = 35;  // IN3 R3
  [1-30:u32:404361] = 36;  // IN4 R4
  [1-30:u32:404363] = 37;  // IN5 R5
  [1-30:u32:404365] = 18;  // IN6 STOP
  [1-30:u32:404367] = 39;  // IN7 R7

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD64150] = [1-30:s32:400199];
  [1-30:s32:400909] = [s32:GD64150];

  // Configuration の実行
  [1-30:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD64156] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD64156] > 1 && [b:GB62314] == 1){

  [1-30:s32:400395] = 1;           // P-PRESET(プリセット)実行する
  [1-30:s32:400909] = [s32:GD64152]; // バックアップしたプリセット位置パラメータを元に戻す
  [1-30:s32:400397] = 0;           // Configuration 実行のゼロクリア
  [1-30:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62313] = 0;              // 運転準備開始トリガのクリア
                           // (このスクリプトの起動トリガを終了する)
  [b:GB62314] = 0;              // 運転準備中フラグのクリア
  [b:GB62315] = 1;              // タッチパネル運転中フラグ(1:運転中)
}

// 31 軸目
// 運転準備開始トリガ かつ 運転準備が完了していない場合
if([b:GB62323] == 1 && [b:GB62324] == 0){

  [b:GB62324] = 1;                     // 運転準備中フラグ(1:運転準備中)
  bmov([1-31:u32:404449],[u32:GD64202],16);    // NET-IN0～15 のバックアップ
  bmov([1-31:u32:404353],[u32:GD64234],8);     // IN0～7 のバックアップ
  [s32:GD64252] = [1-31:s32:400909];      // プリセット位置のバックアップ
  [u32:GD64254] = [1-31:u32:404169];      // JOG 移動量のバックアップ(GOT の最小移動量に使用)

  // NET-IN の入力値をクリアする
  [1-31:w:400126] = 0;

  // タッチパネルから運転する為に信号の割当を変更する
  [1-31:u32:404449] = 48;    // NET-IN0  M0
  [1-31:u32:404451] = 49;    // NET-IN1  M1
  [1-31:u32:404453] = 50;    // NET-IN2  M2
  [1-31:u32:404455] = 51;    // NET-IN3  M3
```

```
   [1-31:u32:404457] = 52;     // NET-IN4  M4
   [1-31:u32:404459] = 53;     // NET-IN5  M5
   [1-31:u32:404461] = 4;   // NET-IN6  START
   [1-31:u32:404463] = 3;   // NET-IN7  HOME
   [1-31:u32:404465] = 18;     // NET-IN8  STOP
   [1-31:u32:404467] = 9;   // NET-IN9  MS1
   [1-31:u32:404469] = 10;     // NET-IN10 MS2
   [1-31:u32:404471] = 5;   // NET-IN11 SSTART
   [1-31:u32:404473] = 6;   // NET-IN12 +JOG
   [1-31:u32:404475] = 7;   // NET-IN13 -JOG
   [1-31:u32:404477] = 1;   // NET-IN14 FWD
   [1-31:u32:404479] = 2;   // NET-IN15 RVS

   [1-31:u32:404353] = 32;  // IN0 R0
   [1-31:u32:404355] = 33;  // IN1 R1
   [1-31:u32:404357] = 34;  // IN2 R2
   [1-31:u32:404359] = 35;  // IN3 R3
   [1-31:u32:404361] = 36;  // IN4 R4
   [1-31:u32:404363] = 37;  // IN5 R5
   [1-31:u32:404365] = 18;  // IN6 STOP
   [1-31:u32:404367] = 39;  // IN7 R7

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD64250] = [1-31:s32:400199];
   [1-31:s32:400909] = [s32:GD64250];

   // Configuration の実行
   [1-31:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD64256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転準備が完了した場合
if([w:GS7] - [w:GD64256] > 1 && [b:GB62324] == 1){

   [1-31:s32:400395] = 1;           // P-PRESET(プリセット)実行する
   [1-31:s32:400909] = [s32:GD64152]; // バックアップしたプリセット位置パラメータを元に戻す
   [1-31:s32:400397] = 0;           // Configuration 実行のゼロクリア
   [1-31:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62323] = 0;               // 運転準備開始トリガのクリア
                              // (このスクリプトの起動トリガを終了する)
   [b:GB62324] = 0;               // 運転準備中フラグのクリア
   [b:GB62325] = 1;               // タッチパネル運転中フラグ(1:運転中)
}
```

スクリプト No.	31046	スクリプト名	Script31046
コメント	29-31_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62305

```
// タッチパネルから位置決め運転をする為のスクリプト
// 選択した運転データ No.を M0～M5(割付直した NET-IN)に反映させる
// 反映した M0～M5 は、ドライバ入力指令(400126)にて、ON する

// 29-31_マルチ運転中スクリプト No.31046
// 29-31_マルチ運転準備スクリプト(No.31045)の GB62305、GB62315、GB62325 の ON 中で、スクリプトを起
動
```

```
// [b:GB62305], [b:GB62315], [b:GB62325] : タッチパネル運転中フラグ(このスクリプトの起動トリガ)
// また、タッチパネル運転中は、他の画面に遷移させないインターロックも兼ねるビット

// [1-29:u32:400195] ～ [1-31:u32:400195] : モニタの現在の選択データ No.の Modbus アドレス
// [u32:GD64000], [u32GD64100], [GD64200] : 運転データ No.のデバイス

// 選択した運転データ No.をドライバ入力指令の M0～M5 に反映させる
// タッチパネルの処理負荷を軽減させる為に、
// モニタの現在の選択データ No.(400195)と選択した運転データ No.を
// 比較して、異なった時のみ、ドライバ入力指令(400126)に反映させる

// 29 軸目
if([b:GB62305] == 1){

   if([1-29:u32:400195] != [u32:GD64000]){
      [1-29:b:400126.b0] = [b:GD64000.b0];    // M0
      [1-29:b:400126.b1] = [b:GD64000.b1];    // M1
      [1-29:b:400126.b2] = [b:GD64000.b2];    // M2
      [1-29:b:400126.b3] = [b:GD64000.b3];    // M3
      [1-29:b:400126.b4] = [b:GD64000.b4];    // M4
      [1-29:b:400126.b5] = [b:GD64000.b5];    // M5
   }
}

// 30 軸目
if([b:GB62315] == 1){

   if([1-30:u32:400195] != [u32:GD64100]){
      [1-30:b:400126.b0] = [b:GD64100.b0];    // M0
      [1-30:b:400126.b1] = [b:GD64100.b1];    // M1
      [1-30:b:400126.b2] = [b:GD64100.b2];    // M2
      [1-30:b:400126.b3] = [b:GD64100.b3];    // M3
      [1-30:b:400126.b4] = [b:GD64100.b4];    // M4
      [1-30:b:400126.b5] = [b:GD64100.b5];    // M5
   }
}

// 31 軸目
if([b:GB62325] == 1){

   if([1-31:u32:400195] != [u32:GD64200]){
      [1-31:b:400126.b0] = [b:GD64200.b0];    // M0
      [1-31:b:400126.b1] = [b:GD64200.b1];    // M1
      [1-31:b:400126.b2] = [b:GD64200.b2];    // M2
      [1-31:b:400126.b3] = [b:GD64200.b3];    // M3
      [1-31:b:400126.b4] = [b:GD64200.b4];    // M4
      [1-31:b:400126.b5] = [b:GD64200.b5];    // M5
   }
}
```

スクリプト No.	31047	スクリプト名	Script31047
コメント	29-31_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62306

```
// タッチパネルからの運転を終了するスクリプト
// バックアップした NET-IN などの信号を元に戻したり、必要なパラメータなどを元に戻す
```

```
// 29-31 軸のマルチ運転の画面スクリプト

// 29-31_マルチ運転終了スクリプト No.31047
// 29-31_マルチ運転開始_終了前確認スクリプト(No.31044)の GB62306、GB62316、GB62326 の ON 中で、スクリプトを起動

// [b:GB62305],    [b:GB62315],    [b:GB62325]    : タッチパネル運転中フラグ
// [b:GB62306],    [b:GB62316],    [b:GB62326]    : 運転終了開始トリガ(このスクリプトの起動トリガ)
// [b:GB62307],    [b:GB62317],    [b:GB62327]    : 運転終了中フラグ
// [u32:GD64002], [u32:GD64102], [u32:GD64202]  : NET-IN0～15 のバックアップデバイス
// [u32:GD64034], [u32:GD64134], [u32:GD64234]  : IN0～7 のバックアップデバイス
// [s32:GD64050], [s32:GD64150], [s32:GD64250]    : モニタの指令位置のバックアップデバイス
// [s32:GD64052], [s32:GD64152], [s32:GD64252]  : プリセット位置のバックアップデバイス
// [u32:GD64054], [u32:GD64154], [u32:GD64254]  : JOG 移動量のバックアップデバイス(GOT の最小移動量に使用)
// [w:GD64056],   [w:GD64156],   [w:GD64256]     : Configuration 実行待ち時間用タイマー

// 29 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62306] == 1 && [b:GB62307] == 0){

   [b:GB62305] = 0;                      // タッチパネル運転中フラグのクリア
   [b:GB62307] = 1;                      // 運転終了中フラグ(1：運転終了中)
   bmov([u32:GD64002],[1-29:u32:404449],16);    // バックアップした NET-IN0～15 を元に戻す
   bmov([u32:GD64034],[1-29:u32:404353],8);     // バックアップした IN0～7 を元に戻す
   [1-29:u32:404169] = [u32:GD64054];      // バックアップした JOG 移動量を元に戻す

   // プリセット位置パラメータにモニタの指令位置をセットする
   [s32:GD64050] = [1-29:s32:400199];
   [1-29:s32:400909] = [s32:GD64050];

   // NET-IN の入力値をクリアする
   [1-29:w:400126] = 0;

   // Configuration の実行
   [1-29:s32:400397] = 1;

   // Configuration 実行待ち時間のタイマー記録(単位は秒)
   [w:GD64056] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD64056] > 1 && [b:GB62307] == 1){

   [1-29:s32:400395] = 1;           // P-PRESET(プリセット)実行する
   [1-29:s32:400909] = [s32:GD64052]; // プリセット位置パラメータを元に戻す
   [1-29:s32:400397] = 0;           // Configuration 実行のゼロクリア
   [1-29:s32:400395] = 0;           // P-PRESET(プリセット)実行のゼロクリア

   [b:GB62306] = 0;              // 運転終了開始トリガのクリア
                             // (このスクリプトの起動トリガを終了する)
   [b:GB62307] = 0;              // 運転終了中フラグのクリア
}

// 30 軸目
```

```
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62316] == 1 && [b:GB62317] == 0){

  [b:GB62315] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62317] = 1;                    // 運転終了中フラグ(1：運転終了中)
  bmov([u32:GD64102],[1-30:u32:404449],16);   // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD64134],[1-30:u32:404353],8);    // バックアップした IN0～7 を元に戻す
  [1-30:u32:404169] = [u32:GD64154];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD64150] = [1-30:s32:400199];
  [1-30:s32:400909] = [s32:GD64150];

  // NET-IN の入力値をクリアする
  [1-30:w:400126] = 0;

  // Configuration の実行
  [1-30:s32:400397] = 1;

  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD64156] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD64156] > 1 && [b:GB62317] == 1){

  [1-30:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-30:s32:400909] = [s32:GD64152]; // プリセット位置パラメータを元に戻す
  [1-30:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-30:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62316] = 0;              // 運転終了開始トリガのクリア
                            // (このスクリプトの起動トリガを終了する)
  [b:GB62317] = 0;              // 運転終了中フラグのクリア
}

// 31 軸目
// 運転終了開始トリガ かつ 運転終了が完了していない場合
if([b:GB62326] == 1 && [b:GB62327] == 0){

  [b:GB62325] = 0;                    // タッチパネル運転中フラグのクリア
  [b:GB62327] = 1;                    // 運転終了中フラグ(1：運転終了中)
  bmov([u32:GD64202],[1-31:u32:404449],16);   // バックアップした NET-IN0～15 を元に戻す
  bmov([u32:GD64234],[1-31:u32:404353],8);    // バックアップした IN0～7 を元に戻す
  [1-31:u32:404169] = [u32:GD64254];      // バックアップした JOG 移動量を元に戻す

  // プリセット位置パラメータにモニタの指令位置をセットする
  [s32:GD64250] = [1-31:s32:400199];
  [1-31:s32:400909] = [s32:GD64250];

  // NET-IN の入力値をクリアする
  [1-31:w:400126] = 0;

  // Configuration の実行
  [1-31:s32:400397] = 1;
```

```
  // Configuration 実行待ち時間のタイマー記録(単位は秒)
  [w:GD64256] = [w:GS7];
}

// Configuration 実行完了 かつ 運転終了が完了した場合
if([w:GS7] - [w:GD64256] > 1 && [b:GB62327] == 1){

  [1-31:s32:400395] = 1;          // P-PRESET(プリセット)実行する
  [1-31:s32:400909] = [s32:GD64252]; // プリセット位置パラメータを元に戻す
  [1-31:s32:400397] = 0;          // Configuration 実行のゼロクリア
  [1-31:s32:400395] = 0;          // P-PRESET(プリセット)実行のゼロクリア

  [b:GB62326] = 0;              // 運転終了開始トリガのクリア
                              // (このスクリプトの起動トリガを終了する)
  [b:GB62327] = 0;              // 運転終了中フラグのクリア
}
```

スクリプト No.	31044	スクリプト名	Script31044
コメント	29-31_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62310

ベース画面 31029 のスクリプト No.31044 と同じです。

スクリプト No.	31045	スクリプト名	Script31045
コメント	29-31_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62313

ベース画面 31029 のスクリプト No.31045 と同じです。

スクリプト No.	31046	スクリプト名	Script31046
コメント	29-31_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62315

ベース画面 31029 のスクリプト No.31046 と同じです。

スクリプト No.	31047	スクリプト名	Script31047
コメント	29-31_マルチ運転終了		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62316

ベース画面 31029 のスクリプト No.31047 と同じです。

スクリプト No.	31044	スクリプト名	Script31044
コメント	29-31_マルチ運転開始_終了前確認		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 2(秒) GB62320

ベース画面 31029 のスクリプト No.31044 と同じです。

スクリプト No.	31045	スクリプト名	Script31045
コメント	29-31_マルチ運転準備		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62323

ベース画面 31029 のスクリプト No.31045 と同じです。

スクリプト No.	31046	スクリプト名	Script31046
コメント	29-31_マルチ運転中		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62325

ベース画面 31029 のスクリプト No.31046 と同じです。

スクリプト No.	31047	スクリプト名	Script31047

コメント	29-31_マルチ運転終了		
データ形式	符号付きBIN16	トリガ種別	ON中 GB62326

ベース画面31029のスクリプトNo.31047と同じです。

ウィンドウ画面30002

スクリプトNo.	30002	スクリプト名	Script30002
コメント	マニュアル表示画面の言語切り換え		
データ形式	符号付きBIN16	トリガ種別	画面を閉じる時

```
if(([w:GD60000] >= 30500) && ([w:GD60000] <= 30502)){  //ベース画面切り換えデバイス値が 30500～30502の場合
   if([w:GD60021] == 1){                         //言語が言語1の場合
      [w:GD60000] = 30500;                       //マニュアル表示-言語1画面に遷移
   }
   if([w:GD60021] == 2){                         //言語が言語2の場合
      [w:GD60000] = 30501;                       //マニュアル表示-言語2画面に遷移
   }
   if([w:GD60021] == 3){                         //言語が言語3の場合
      [w:GD60000] = 30502;                       //マニュアル表示-言語3画面に遷移
   }
}
```

ウィンドウ画面30004

スクリプトNo.	30100	スクリプト名	Script30100
コメント	軸切り換え		
データ形式	符号なしBIN16	トリガ種別	画面を閉じる時

```
//軸切り換えのデータをダミーデバイスから各デバイスに展開する
if([b:GB61000] == ON){
   [w:GD61000] = [w:GD65000];

   if([w:GD65001] == 1){
      [b:GB61010] = OFF;
   }
   if([w:GD65001] == 2){
      [b:GB61010] = ON;
   }
}

[w:GD65000]=0;
[w:GD65001]=0;
rst([b:GB61000]);
rst([b:GB61001]);
rst([b:GB61002]);
```

ウィンドウ画面 32001

スクリプト No.	31006	スクリプト名	Script31006
コメント	運転データ入力_書込み		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62003

```
// 運転データ入力のオーバーラップウィンドウで、設定した値をドライバに書いて、
// 運転データ画面に表示する

// 運転データ入力のオーバーラップウィンドウ画面スクリプト No.31006
// 上記オーバーラップウィンドウの「設定」ボタン(GB62003)の ON 中で、スクリプトを起動

// [b:GB62003] ： 本スクリプトの起動トリガ、ビットセットで起動(「設定」ボタン)
// [w:GD61011] ： Modbus アドレスのオフセット値
// [w:GD61012] ： 運転データ No.のタッチ位置情報(何行目)

// タッチされた行に、運転データを反映させる(ドライバへの書込みと運転データ画面に反映させる)
switch ([w:GD61012])
{
   case 0 : [u16:401282[w:GD61011]]= [u16:GD61021];    // 0 行目の運転方式
          [s32:401025[w:GD61011]] = [s32:GD61022];    // 0 行目の位置
          [u32:401153[w:GD61011]] = [u32:GD61024];    // 0 行目の運転速度
          [u16:401410[w:GD61011]] = [u16:GD61026];    // 0 行目の運転機能
          [u32:401793[w:GD61011]] = [u32:GD61027];    // 0 行目の押し当て電流
          [u32:402049[w:GD61011]] = [u32:GD61029];    // 0 行目のドウェル時間
          [u16:401922[w:GD61011]] = [u16:GD61031];    // 0 行目の順送り位置決め運転
          [u32:401537[w:GD61011]] = [u32:GD61032];    // 0 行目の加速
          [u32:401665[w:GD61011]] = [u32:GD61034];    // 0 行目の減速
          break;

   case 1 : [u16:401284[w:GD61011]]= [u16:GD61021];    // 1 行目の運転方式
          [s32:401027[w:GD61011]] = [s32:GD61022];    // 1 行目の位置
          [u32:401155[w:GD61011]] = [u32:GD61024];    // 1 行目の運転速度
          [u16:401412[w:GD61011]] = [u16:GD61026];    // 1 行目の運転機能
          [u32:401795[w:GD61011]] = [u32:GD61027];    // 1 行目の押し当て電流
          [u32:402051[w:GD61011]] = [u32:GD61029];    // 1 行目のドウェル時間
          [u16:401924[w:GD61011]] = [u16:GD61031];    // 1 行目の順送り位置決め運転
          [u32:401539[w:GD61011]] = [u32:GD61032];    // 1 行目の加速
          [u32:401667[w:GD61011]] = [u32:GD61034];    // 1 行目の減速
          break;

   case 2 : [u16:401286[w:GD61011]]= [u16:GD61021];    // 2 行目の運転方式
          [s32:401029[w:GD61011]] = [s32:GD61022];    // 2 行目の位置
          [u32:401157[w:GD61011]] = [u32:GD61024];    // 2 行目の運転速度
          [u16:401414[w:GD61011]] = [u16:GD61026];    // 2 行目の運転機能
          [u32:401797[w:GD61011]] = [u32:GD61027];    // 2 行目の押し当て電流
          [u32:402053[w:GD61011]] = [u32:GD61029];    // 2 行目のドウェル時間
          [u16:401926[w:GD61011]] = [u16:GD61031];    // 2 行目の順送り位置決め運転
          [u32:401541[w:GD61011]] = [u32:GD61032];    // 2 行目の加速
          [u32:401669[w:GD61011]] = [u32:GD61034];    // 2 行目の減速
          break;

   case 3 : [u16:401288[w:GD61011]]= [u16:GD61021];    // 3 行目の運転方式
          [s32:401031[w:GD61011]] = [s32:GD61022];    // 3 行目の位置
          [u32:401159[w:GD61011]] = [u32:GD61024];    // 3 行目の運転速度
          [u16:401416[w:GD61011]] = [u16:GD61026];    // 3 行目の運転機能
          [u32:401799[w:GD61011]] = [u32:GD61027];    // 3 行目の押し当て電流
```

```
        [u32:402055[w:GD61011]] = [u32:GD61029];    // 3 行目のドウェル時間
        [u16:401928[w:GD61011]] = [u16:GD61031];    // 3 行目の順送り位置決め運転
        [u32:401543[w:GD61011]] = [u32:GD61032];    // 3 行目の加速
        [u32:401671[w:GD61011]] = [u32:GD61034];    // 3 行目の減速
        break;

  case 4 : [u16:401290[w:GD61011]]= [u16:GD61021];    // 4 行目の運転方式
        [s32:401033[w:GD61011]] = [s32:GD61022];    // 4 行目の位置
        [u32:401161[w:GD61011]] = [u32:GD61024];    // 4 行目の運転速度
        [u16:401418[w:GD61011]] = [u16:GD61026];    // 4 行目の運転機能
        [u32:401801[w:GD61011]] = [u32:GD61027];    // 4 行目の押し当て電流
        [u32:402057[w:GD61011]] = [u32:GD61029];    // 4 行目のドウェル時間
        [u16:401930[w:GD61011]] = [u16:GD61031];    // 4 行目の順送り位置決め運転
        [u32:401545[w:GD61011]] = [u32:GD61032];    // 4 行目の加速
        [u32:401673[w:GD61011]] = [u32:GD61034];    // 4 行目の減速
        break;

  case 5 : [u16:401292[w:GD61011]]= [u16:GD61021];    // 5 行目の運転方式
        [s32:401035[w:GD61011]] = [s32:GD61022];    // 5 行目の位置
        [u32:401163[w:GD61011]] = [u32:GD61024];    // 5 行目の運転速度
        [u16:401420[w:GD61011]] = [u16:GD61026];    // 5 行目の運転機能
        [u32:401803[w:GD61011]] = [u32:GD61027];    // 5 行目の押し当て電流
        [u32:402059[w:GD61011]] = [u32:GD61029];    // 5 行目のドウェル時間
        [u16:401932[w:GD61011]] = [u16:GD61031];    // 5 行目の順送り位置決め運転
        [u32:401547[w:GD61011]] = [u32:GD61032];    // 5 行目の加速
        [u32:401675[w:GD61011]] = [u32:GD61034];    // 5 行目の減速
        break;

  case 6 : [u16:401294[w:GD61011]]= [u16:GD61021];    // 6 行目の運転方式
        [s32:401037[w:GD61011]] = [s32:GD61022];    // 6 行目の位置
        [u32:401165[w:GD61011]] = [u32:GD61024];    // 6 行目の運転速度
        [u16:401422[w:GD61011]] = [u16:GD61026];    // 6 行目の運転機能
        [u32:401805[w:GD61011]] = [u32:GD61027];    // 6 行目の押し当て電流
        [u32:402061[w:GD61011]] = [u32:GD61029];    // 6 行目のドウェル時間
        [u16:401934[w:GD61011]] = [u16:GD61031];    // 6 行目の順送り位置決め運転
        [u32:401549[w:GD61011]] = [u32:GD61032];    // 6 行目の加速
        [u32:401677[w:GD61011]] = [u32:GD61034];    // 6 行目の減速
        break;

  case 7 : [u16:401296[w:GD61011]]= [u16:GD61021];    // 7 行目の運転方式
        [s32:401039[w:GD61011]] = [s32:GD61022];    // 7 行目の位置
        [u32:401167[w:GD61011]] = [u32:GD61024];    // 7 行目の運転速度
        [u16:401424[w:GD61011]] = [u16:GD61026];    // 7 行目の運転機能
        [u32:401807[w:GD61011]] = [u32:GD61027];    // 7 行目の押し当て電流
        [u32:402063[w:GD61011]] = [u32:GD61029];    // 7 行目のドウェル時間
        [u16:401936[w:GD61011]] = [u16:GD61031];    // 7 行目の順送り位置決め運転
        [u32:401551[w:GD61011]] = [u32:GD61032];    // 7 行目の加速
        [u32:401679[w:GD61011]] = [u32:GD61034];    // 7 行目の減速
        break;
}

[b:GB62003] = 0; // このスクリプトの起動トリガを終了する
[w:GD60004] = 0; // オーバーラップウィンドウの終了
```

ウィンドウ画面 32004～32006

スクリプト No.	31007	スクリプト名	Script31007
コメント	IN_OUT 機能選択_書込み		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB62004

```
// IN、OUT、NET-IN、NET-OUT の機能選択をドライバに書き込むスクリプト

// IN 入力機能選択、NET-IN 入力機能選択、OUT/NET-OUT 出力機能選択の
// それぞれのオーバーラップウィンドウ画面スクリプト No.31007
// それぞれの「設定」ボタン(GB32004)の ON 中で、スクリプトを起動

// [b:GB62004] : 本スクリプトの起動トリガ、ビットセットで起動(「設定」ボタン)
// [b:GB62005] : OUT と NET-OUT と区別するビット(1:OUT、0:NET-OUT)
// [u16:GD61040] : IN 入力の信号番号
// [u16:GD61041] : OUT 出力の信号番号
// [u16:GD61042] : NET-IN 入力の信号番号
// [u16:GD61043] : NET-OUT 出力の信号番号

// [u16:GD61044] : IN 入力機能選択の一時保存デバイス
// [u16:GD61046] : OUT/NET-OUT 出力機能選択の一時保存デバイス
// [u16:GD61048] : NET-IN 入力機能選択の一時保存デバイス

// IN 入力機能選択時
if([w:GD60004] == 32004){        // IN 入力機能選択ウィンド表示中の場合
  switch([u16:GD61040]){
    case 0: [u16:404354] = [u16:GD61044];   // IN0 入力機能選択の設定
        break;
    case 1: [u16:404356] = [u16:GD61044];   // IN1 入力機能選択の設定
        break;
    case 2: [u16:404358] = [u16:GD61044];   // IN2 入力機能選択の設定
        break;
    case 3: [u16:404360] = [u16:GD61044];   // IN3 入力機能選択の設定
        break;
    case 4: [u16:404362] = [u16:GD61044];   // IN4 入力機能選択の設定
        break;
    case 5: [u16:404364] = [u16:GD61044];   // IN5 入力機能選択の設定
        break;
    case 6: [u16:404366] = [u16:GD61044];   // IN6 入力機能選択の設定
        break;
    case 7: [u16:404368] = [u16:GD61044];   // IN7 入力機能選択の設定
        break;
  }
}

// NET-IN 入力機能選択時
if([w:GD60004] == 32005){        // NET-IN 入力機能選択ウィンド表示中の場合
  switch([u16:GD61042]){
    case 0 : [u16:404450] = [u16:GD61048]; // NET-IN0 入力機能選択の設定
          break;
    case 1 : [u16:404452] = [u16:GD61048]; // NET-IN1 入力機能選択の設定
          break;
    case 2 : [u16:404454] = [u16:GD61048]; // NET-IN2 入力機能選択の設定
          break;
    case 3 : [u16:404456] = [u16:GD61048]; // NET-IN3 入力機能選択の設定
          break;
    case 4 : [u16:404458] = [u16:GD61048]; // NET-IN4 入力機能選択の設定
```

```
            break;
      case 5 : [u16:404460] = [u16:GD61048]; // NET-IN5 入力機能選択の設定
            break;
      case 6 : [u16:404462] = [u16:GD61048]; // NET-IN6 入力機能選択の設定
            break;
      case 7 : [u16:404464] = [u16:GD61048]; // NET-IN7 入力機能選択の設定
            break;
      case 8 : [u16:404466] = [u16:GD61048]; // NET-IN8 入力機能選択の設定
            break;
      case 9 : [u16:404468] = [u16:GD61048]; // NET-IN9 入力機能選択の設定
            break;
      case 10: [u16:404470] = [u16:GD61048]; // NET-IN10 入力機能選択の設定
            break;
      case 11: [u16:404472] = [u16:GD61048]; // NET-IN11 入力機能選択の設定
            break;
      case 12: [u16:404474] = [u16:GD61048]; // NET-IN12 入力機能選択の設定
            break;
      case 13: [u16:404476] = [u16:GD61048]; // NET-IN13 入力機能選択の設定
            break;
      case 14: [u16:404478] = [u16:GD61048]; // NET-IN14 入力機能選択の設定
            break;
      case 15: [u16:404480] = [u16:GD61048]; // NET-IN15 入力機能選択の設定
            break;
   }
}

// OUT/NET-OUT 出力機能選択時
if([w:GD60004] == 32006){        // OUT/NET-OUT 出力機能選択ウィンド表示中の場合
   if([b:GB62005] == 1){       // OUT 出力機能選択時
      switch([u16:GD61041]){
      case 0:[u16:404418] = [u16:GD61046];    // OUT0 出力機能選択の設定
          break;
      case 1: [u16:404420] = [u16:GD61046];   // OUT1 出力機能選択の設定
          break;
      case 2: [u16:404422] = [u16:GD61046];   // OUT2 出力機能選択の設定
          break;
      case 3: [u16:404424] = [u16:GD61046];   // OUT3 出力機能選択の設定
          break;
      case 4: [u16:404426] = [u16:GD61046];   // OUT4 出力機能選択の設定
          break;
      case 5: [u16:404428] = [u16:GD61046];   // OUT5 出力機能選択の設定
          break;
      }
   } else {                  // NET-OUT 出力機能選択時
      switch([u16:GD61043]){
      case 0 : [u16:404482] = [u16:GD61046]; // NET-OUT0 出力機能選択の設定
            break;
      case 1 : [u16:404484] = [u16:GD61046]; // NET-OUT1 出力機能選択の設定
              break;
      case 2 : [u16:404486] = [u16:GD61046]; // NET-OUT2 出力機能選択の設定
            break;
      case 3 : [u16:404488] = [u16:GD61046]; // NET-OUT3 出力機能選択の設定
            break;
      case 4 : [u16:404490] = [u16:GD61046]; // NET-OUT4 出力機能選択の設定
            break;
      case 5 : [u16:404492] = [u16:GD61046]; // NET-OUT5 出力機能選択の設定
```

```
            break;
      case 6 : [u16:404494] = [u16:GD61046]; // NET-OUT6 出力機能選択の設定
            break;
      case 7 : [u16:404496] = [u16:GD61046]; // NET-OUT7 出力機能選択の設定
            break;
      case 8 : [u16:404498] = [u16:GD61046]; // NET-OUT8 出力機能選択の設定
            break;
      case 9 : [u16:404500] = [u16:GD61046]; // NET-OUT9 出力機能選択の設定
            break;
      case 10: [u16:404502] = [u16:GD61046]; // NET-OUT10 出力機能選択の設定
            break;
      case 11: [u16:404504] = [u16:GD61046]; // NET-OUT11 出力機能選択の設定
            break;
      case 12: [u16:404506] = [u16:GD61046]; // NET-OUT12 出力機能選択の設定
            break;
      case 13: [u16:404508] = [u16:GD61046]; // NET-OUT13 出力機能選択の設定
            break;
      case 14: [u16:404510] = [u16:GD61046]; // NET-OUT14 出力機能選択の設定
            break;
      case 15: [u16:404512] = [u16:GD61046]; // NET-OUT15 出力機能選択の設定
            break;
      }
   }
}

[b:GB62004] = 0; // このスクリプトの起動トリガを終了する
[w:GD60004] = 0; // オーバーラップウィンドウの終了
```

5.6.3 オブジェクトスクリプト

ウィンドウ画面 30003

オブジェクト(名称)	数値表示(年)		
スクリプトユーザ ID	1		
データ形式	符号なし BIN16	トリガ種別	立上り　GB40

```
//時計データより本日の年月を取得
[w:TMP950] = [w:GS650] & 0xF000;//設定用時計データより年の下 2 桁の 10 の位を取得
[w:TMP960] = [w:TMP950] >> 12;//桁合せ
[w:TMP968] = [w:TMP960] * 10;//BCD->BIN
[w:TMP951] = [w:GS650] & 0x0F00;//設定用時計データより年の下 2 桁の 1 の位を取得
[w:TMP961] = [w:TMP951] >> 8;//BCD->BIN
[w:TMP973] = 2000 + [w:TMP968] +  [w:TMP961];//TMP973 に年を BIN でセット
[w:GD64990] = [w:TMP973];//年をセット

[w:TMP952] = [w:GS650] & 0x00F0;//設定用時計データより月の 10 の位を取得
[w:TMP962] = [w:TMP952] >> 4;//桁合せ
[w:TMP969] = [w:TMP962] * 10;//BCD->BIN
[w:TMP953] = [w:GS650] & 0x000F;//設定用時計データより月の 1 の位を取得
[w:TMP974] = [w:TMP969] +  [w:TMP953];//TMP974 に月を BIN でセット
[w:GD64991] = [w:TMP974];//月をセット

[w:TMP954] = [w:GS651] & 0xF000;//設定用時計データより日の下 2 桁の 10 の位を取得
[w:TMP963] = [w:TMP954] >> 12;//桁合せ
[w:TMP970] = [w:TMP963] * 10;//BCD->BIN
[w:TMP955] = [w:GS651] & 0x0F00;//設定用時計データより日の下 2 桁の 1 の位を取得
[w:TMP964] = [w:TMP955] >> 8;//BCD->BIN
[w:TMP975] =[w:TMP970] +  [w:TMP964];//TMP975 に日を BIN でセット
[w:GD64992] = [w:TMP975];//日をセット

[w:TMP956] = [w:GS651] & 0x00F0;//設定用時計データより時の 10 の位を取得
[w:TMP965] = [w:TMP956] >> 4;//桁合せ
[w:TMP971] = [w:TMP965] * 10;//BCD->BIN
[w:TMP957] = [w:GS651] & 0x000F;//設定用時計データより時の 1 の位を取得
[w:TMP976] = [w:TMP971] +  [w:TMP957];//TMP976 に時を BIN でセット
[w:GD64993] = [w:TMP976];//時をセット

[w:TMP958] = [w:GS652] & 0xF000;//設定用時計データより分の下 2 桁の 10 の位を取得
[w:TMP966] = [w:TMP958] >> 12;//桁合せ
[w:TMP972] = [w:TMP966] * 10;//BCD->BIN
[w:TMP959] = [w:GS652] & 0x0F00;//設定用時計データより分の下 2 桁の 1 の位を取得
[w:TMP967] = [w:TMP959] >> 8;//BCD->BIN
[w:TMP977] =[w:TMP972] +  [w:TMP967];//TMP977 に分を BIN でセット
[w:GD64994] = [w:TMP977];//分をセット

[w:TMP993] = [w:GS652] & 0x00F0;//設定用時計データより秒の 10 の位を取得
[w:TMP995] = [w:TMP993] >> 4;//桁合せ
[w:TMP996] = [w:TMP995] * 10;//BCD->BIN
[w:TMP994] = [w:GS652] & 0x000F;//設定用時計データより秒の 1 の位を取得
[w:TMP978] = [w:TMP996] +  [w:TMP994];//TMP978 に秒を BIN でセット
[w:GD64995] = [w:TMP978];//秒をセット
```

オブジェクト(名称)	数値表示(月)		
スクリプトユーザ ID	2		
データ形式	符号なし BIN16	トリガ種別	常時

```
//BIN -> BCD 変換

[w:TMP979] = [w:GD64990] - 2000;// 年の下 2 桁

[w:TMP980] = (([w:TMP979] / 10) << 4) + ([w:TMP979] % 10);  //年 BIN -> BCD
[w:TMP981] = (([w:GD64991] / 10) << 4) + ([w:GD64991] % 10);  //月 BIN -> BCD
[w:TMP982] = (([w:GD64992] / 10) << 4) + ([w:GD64992] % 10);  //日 BIN -> BCD
[w:TMP983] = (([w:GD64993] / 10) << 4) + ([w:GD64993] % 10);  //時 BIN -> BCD
[w:TMP984] = (([w:GD64994] / 10) << 4) + ([w:GD64994] % 10);  //分 BIN -> BCD
[w:TMP985] = (([w:GD64995] / 10) << 4) + ([w:GD64995] % 10);  //秒 BIN -> BCD
```

オブジェクト(名称)	数値表示(日)		
スクリプトユーザ ID	3		
データ形式	符号なし BIN16	トリガ種別	常時

```
// 年月設定

[w:GS513] = ([w:TMP980] << 8) + [w:TMP981];  // 変更時刻デバイスに年月セット
```

オブジェクト(名称)	数値表示(時)		
スクリプトユーザ ID	4		
データ形式	符号なし BIN16	トリガ種別	常時

```
// 日時設定

[w:GS514] = ([w:TMP982] << 8) + [w:TMP983];  // 変更時刻デバイスに日時セット
```

オブジェクト(名称)	数値表示(分)		
スクリプトユーザ ID	5		
データ形式	符号なし BIN16	トリガ種別	常時

```
// 分秒設定

[w:GS515] = ([w:TMP984] << 8) + [w:TMP985];  // 変更時刻デバイスに分秒セット
```

オブジェクト(名称)	数値表示(秒)		
スクリプトユーザ ID	6		
データ形式	符号なし BIN16	トリガ種別	常時

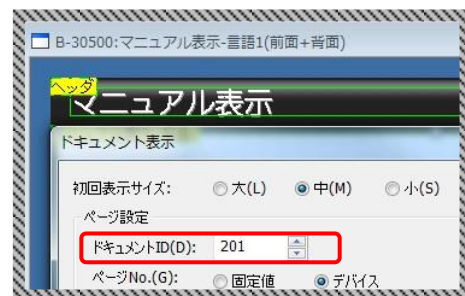
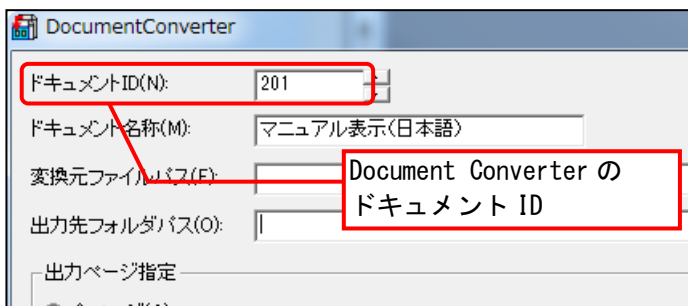
```
// 曜日設定

[w:TMP986] = [w:GD63990];  //年(BIN)
[w:TMP987] = [w:GD63991];  //月(BIN)
[w:TMP988] = [w:GD63992];  //日(BIN)

if(([w:TMP987] == 1) || ([w:TMP987] == 2)){//1･2 月の場合のみ前年の 13･14 月として計算するための補正
処理
   [w:TMP986] =[w:TMP986] - 1; //年から 1 を減算
   [w:TMP987] =[w:TMP987] + 12;//月に 12 を加算
}

[w:TMP989] = [w:TMP986]/4;//ツェラーの公式に必要な項を作成
[w:TMP990] = [w:TMP986]/100;//ツェラーの公式に必要な項を作成
[w:TMP991] = [w:TMP986]/400;//ツェラーの公式に必要な項を作成
[w:TMP992] = (13*[w:TMP987]+8)/5;//ツェラーの公式に必要な項を作成

//ツェラーの公式で曜日算出して変更時刻デバイスに曜日をセット
[w:GS516] = ([w:TMP986]+[w:TMP989]-[w:TMP990]+[w:TMP991]+[w:TMP992]+[w:TMP988])%7;
```

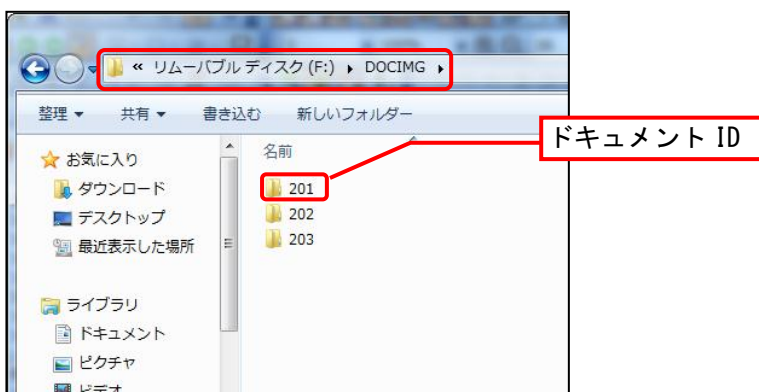
6. マニュアル表示について

マニュアル表示は、ドキュメント表示機能を使用して表示しています。ドキュメント表示機能の詳細については、「GT Designer3 (GOT2000) ヘルプ」を参照してください。なお、ドキュメント表示機能は、言語切り換えに非対応のため、サンプル画面では、言語切り換え時に選択した言語のドキュメント(ドキュメント ID)を設定したベース画面を切り換え表示することで言語切り換えを実現しています。

6.1 マニュアル表示用ドキュメントデータの準備

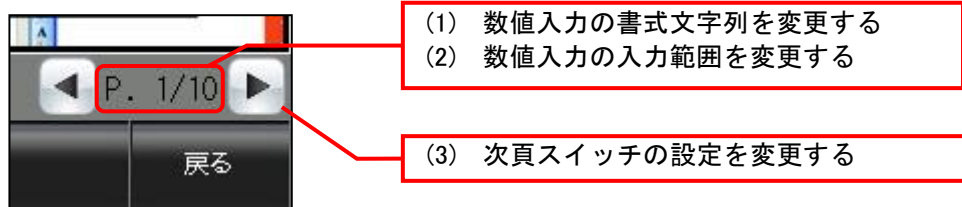
例:ベース画面 B-30500:マニュアル表示-言語1にマニュアル(ドキュメント)を表示する場合

(1) 表示するマニュアル(Word や Excel など)を Document Converter を使用してドキュメント表示機能用のドキュメントデータ(JPEG ファイル)に変換します。この際、Document Converter の[ドキュメント ID]には、ベース画面 B-30500 のドキュメント表示の[ドキュメント ID]と同じ値を設定します。

例:ベース画面 B-30500:マニュアル表示-言語1のドキュメント表示のドキュメント ID

(2) ドキュメントデータは DOCIMG フォルダの 201 フォルダ内に生成されます。DOCIMG フォルダ以下のフォルダ構成は変更せずに、DOCIMG フォルダごと SD カードのルートディレクトリに保存してください。

SD カードのフォルダ構成

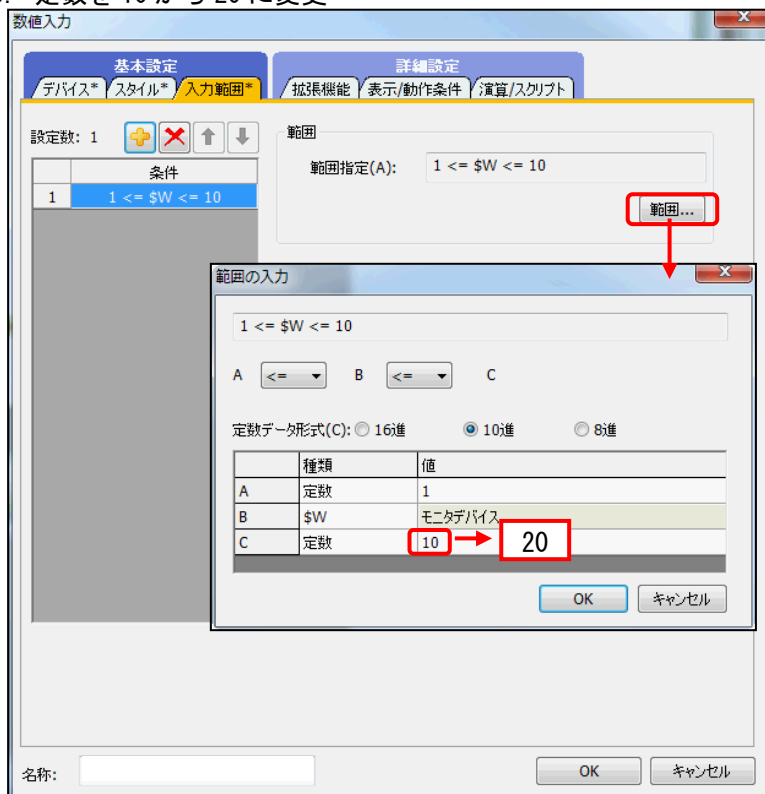
6.2 ドキュメントの総ページ数の変更

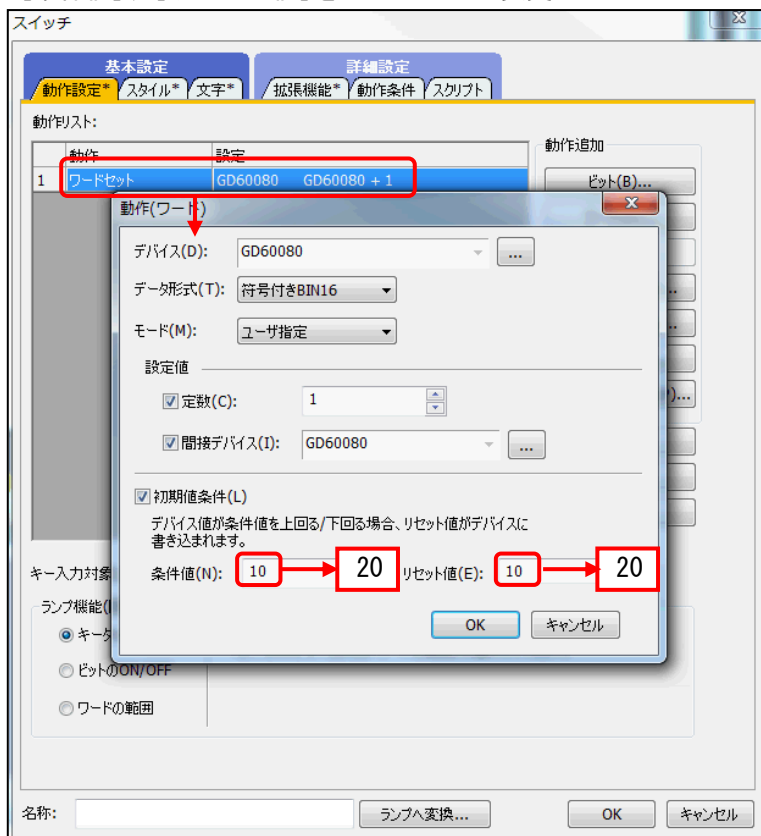
表示するドキュメントの総ページ数に合わせて、画面右下に表示する総ページ数を変更してください。

例：ドキュメント総数を 10 ページから 20 ページに変更する場合

(1) 数値入力の書式を変更する。

1. 数値入力をダブルクリックし、ダイアログボックスの[デバイス]タブを表示
2. [書式文字列]を「P.##/10」から「P.##/20」に変更

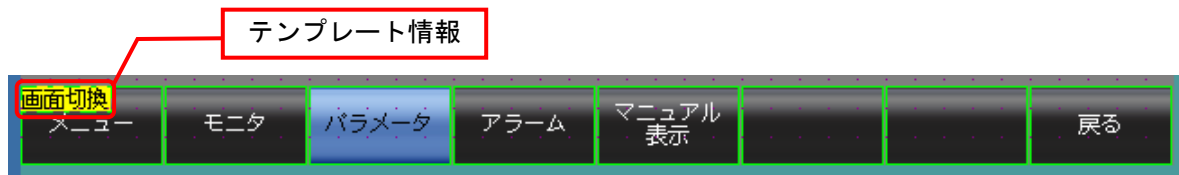
(2)　数値入力の入力範囲を変更する。

1.　ダイアログボックスの[入力範囲]タブを表示
2.　[範囲]をクリックし、[範囲の入力]ダイアログボックスを表示
3.　定数を 10 から 20 に変更

(3)　次頁スイッチの設定を変更する

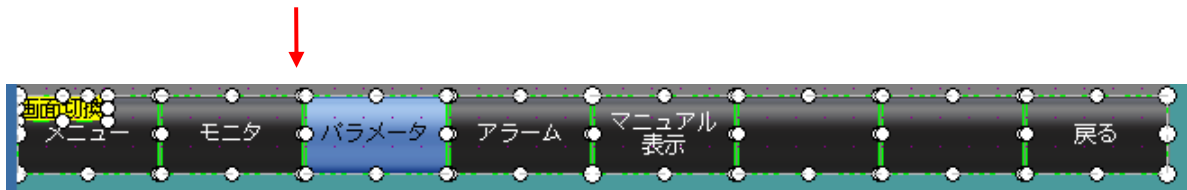
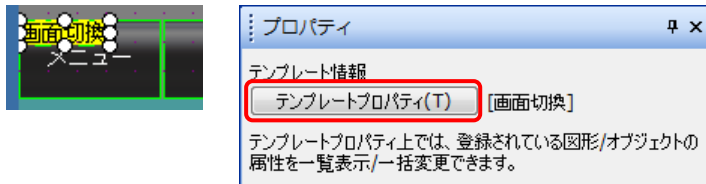
1.　ダイアログボックスの[動作設定]タブを表示
2.　[動作 1]をダブルクリックし、[動作(ワード)]ダイアログボックスを表示
3.　[条件値]、[リセット値]を 10 から 20 に変更

6.3 「マニュアル表示」スイッチの設定

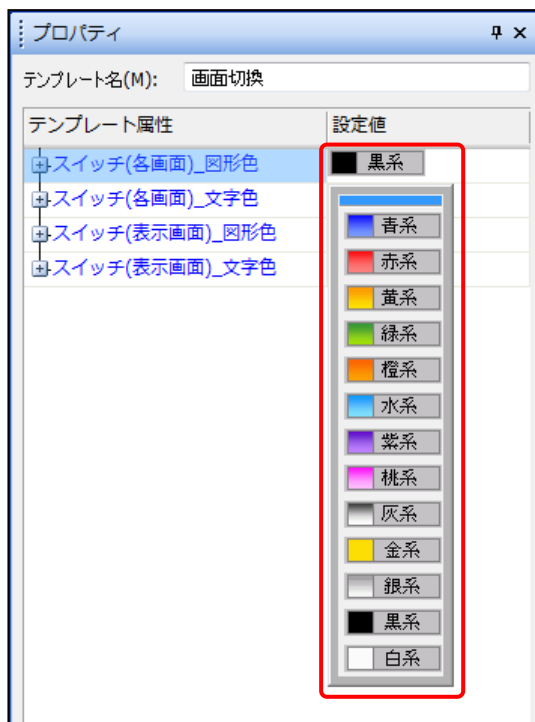
「マニュアル表示」スイッチは、言語切り換えデバイスに格納された列 No. によって表示するマニュアル画面を指定しています。列 No. の詳細については、「5.1　表示言語」を参照してください。

7. テンプレート

テンプレートとは、図形やオブジェクトの集合体です。関連のある設定をテンプレート属性としてまとめて登録しているためデバイスや色などを簡単に一括変更できます。属性の設定値を変更する詳細については、「GT Designer3 (GOT2000) ヘルプ」を参照してください。

テンプレート情報は作画ソフトウェアの編集画面上にのみ表示され、GOT の表示画面上には表示されません。

例：スイッチ(各画面)の図形色を変更する場合

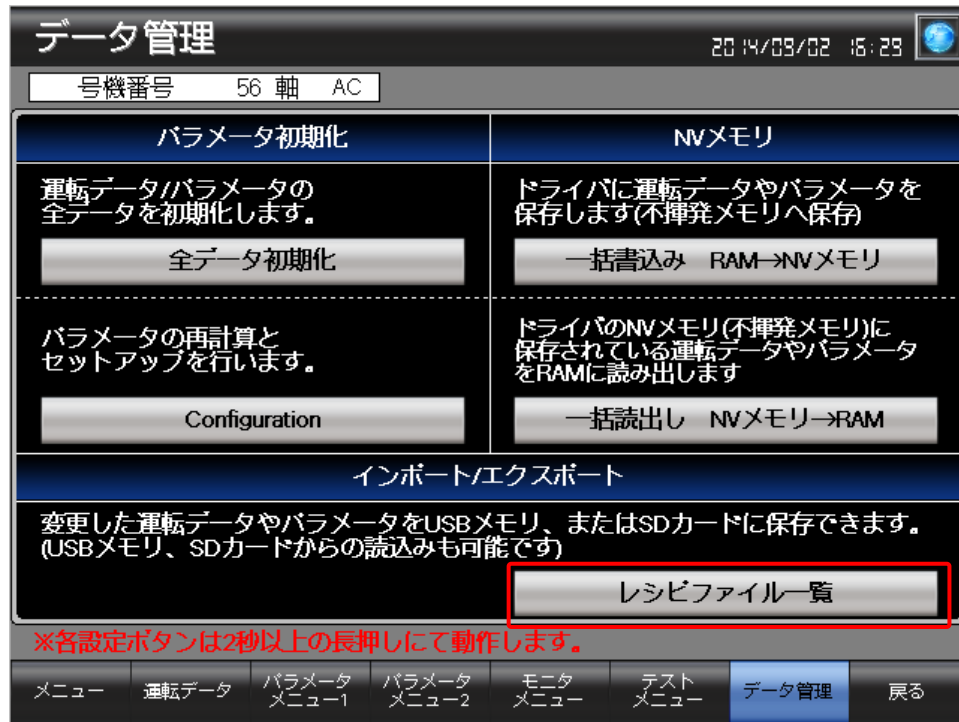
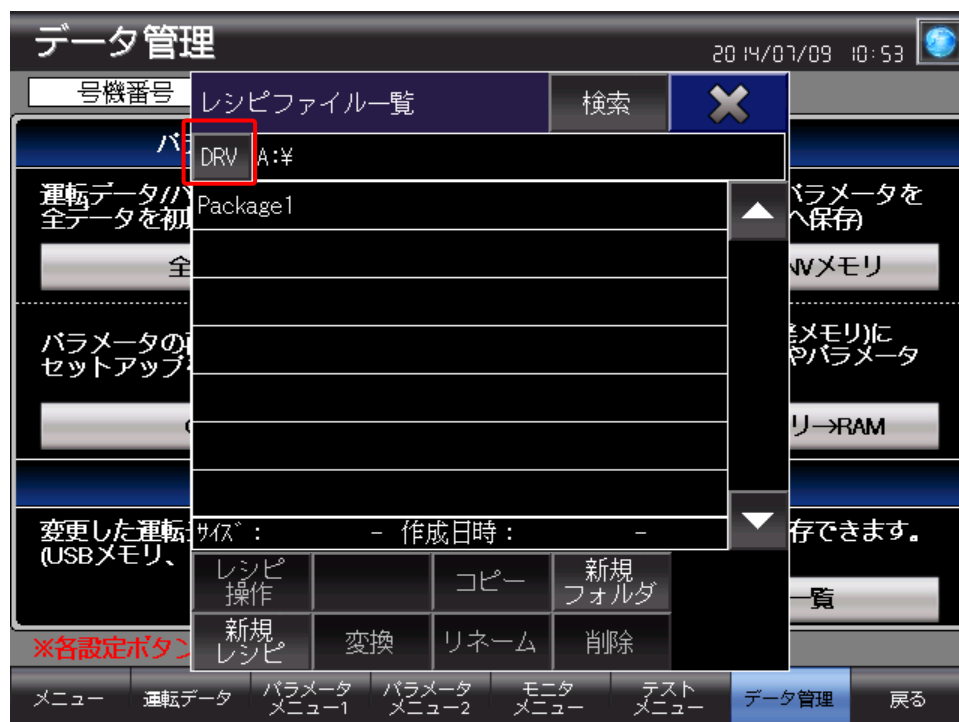
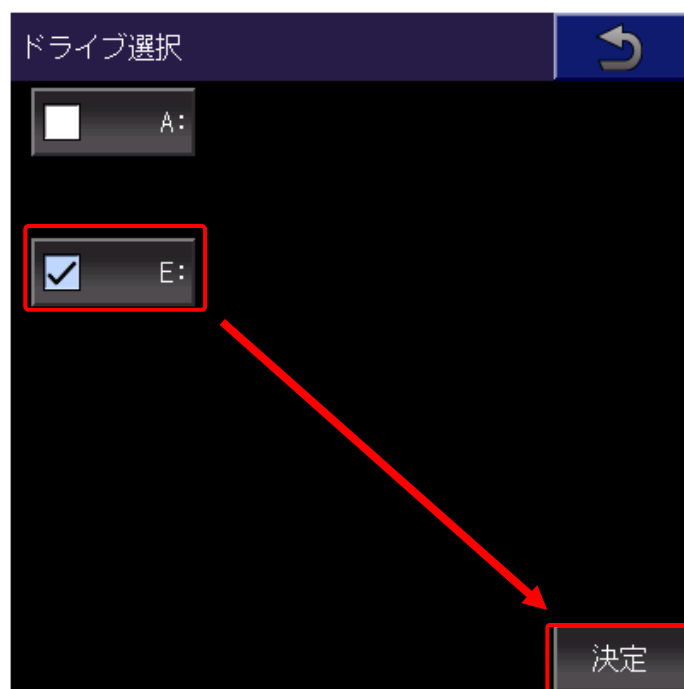
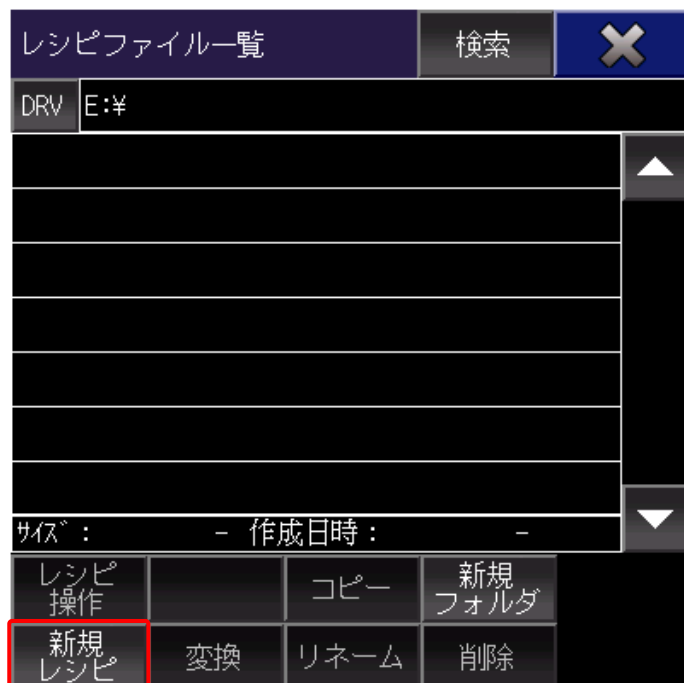
(1) [テンプレート情報]を選択し[テンプレートプロパティ]をクリック(または[テンプレート情報]をダブルクリック)

テンプレートに登録されている図形やオブジェクトが選択状態になります。

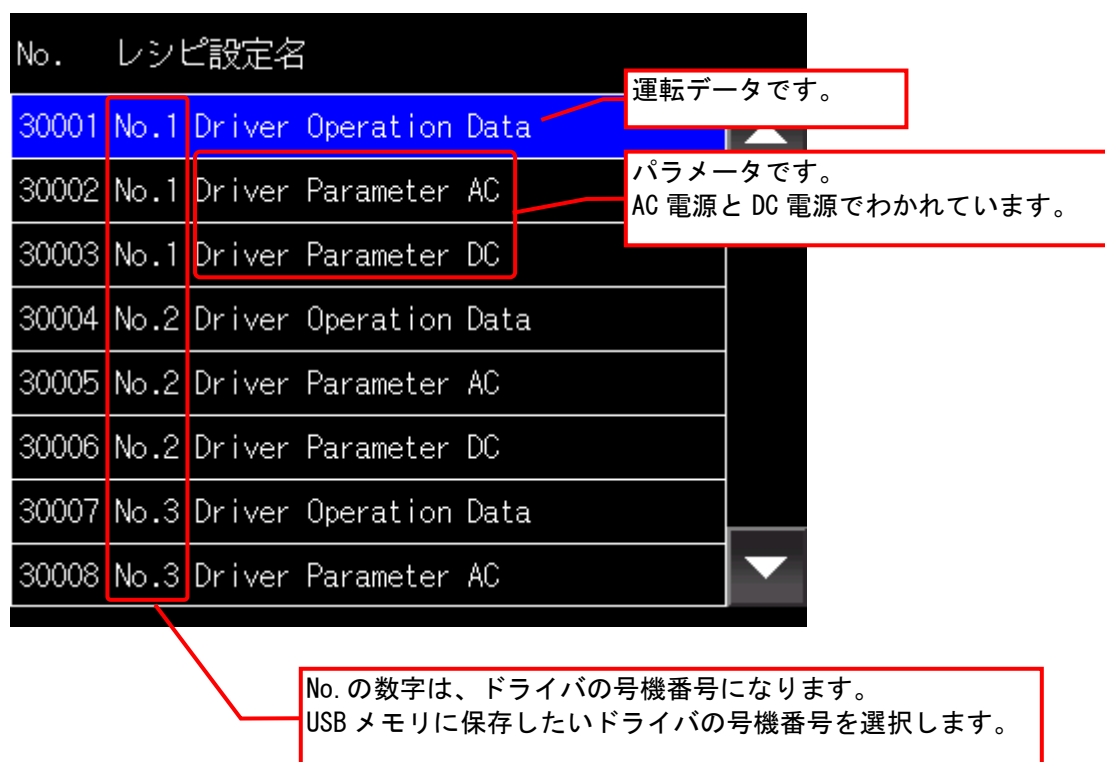
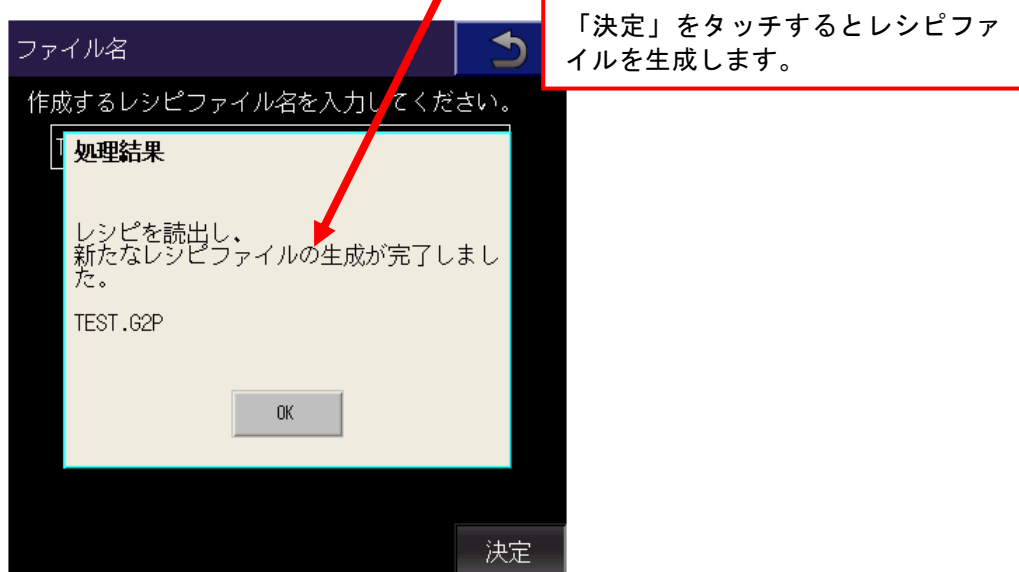
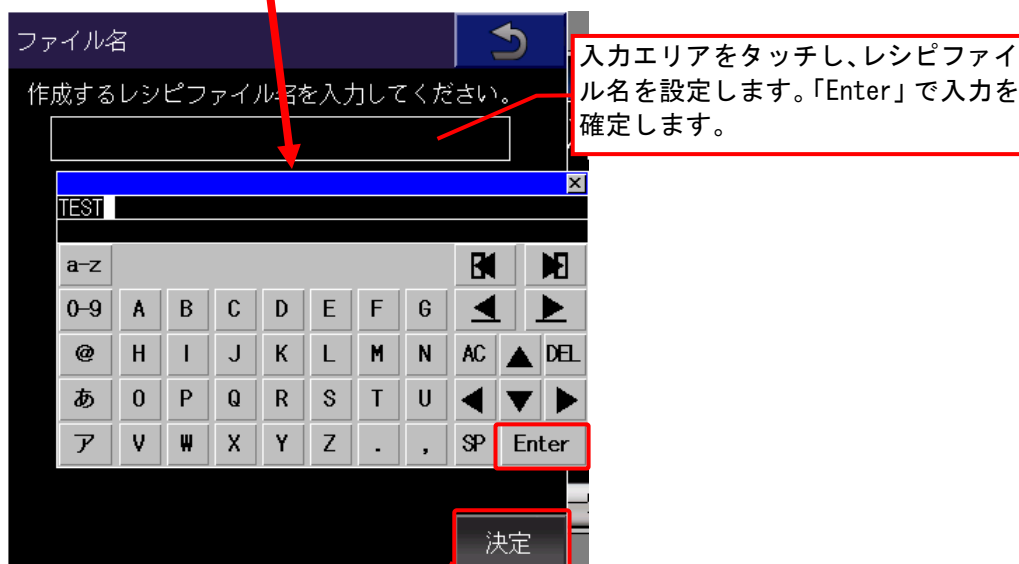
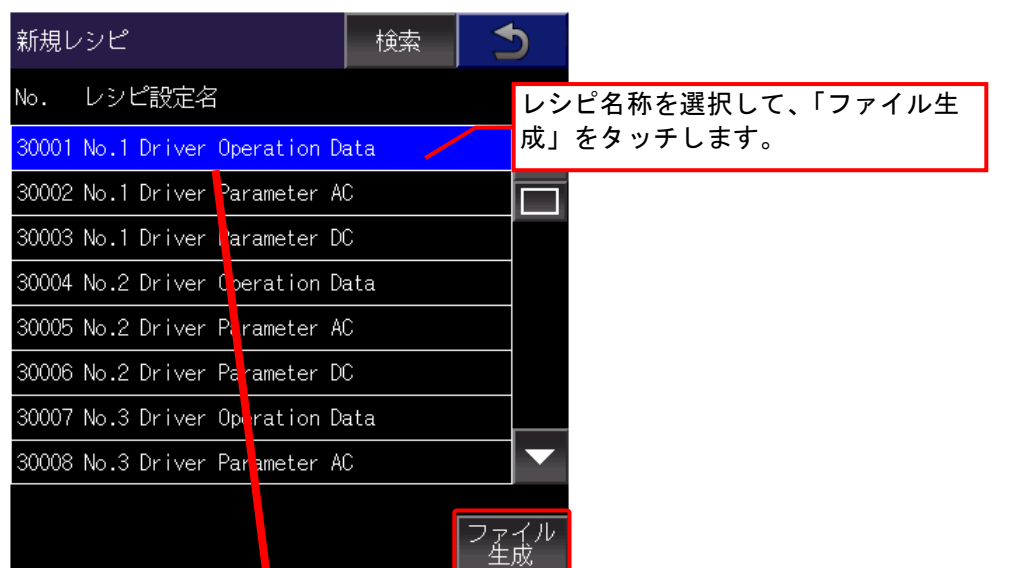
(2) [スイッチ(各画面)_図形色]の[設定値]をダブルクリックして、変更したい色を選択

8. 運転データやパラメータの USB メモリ、SD カードへの保存・読出しについて

レシピファイルを使用することで、運転データやパラメータを USB メモリや SD カードへ保存したり、USB メモリや SD カードから読出しすることができます。

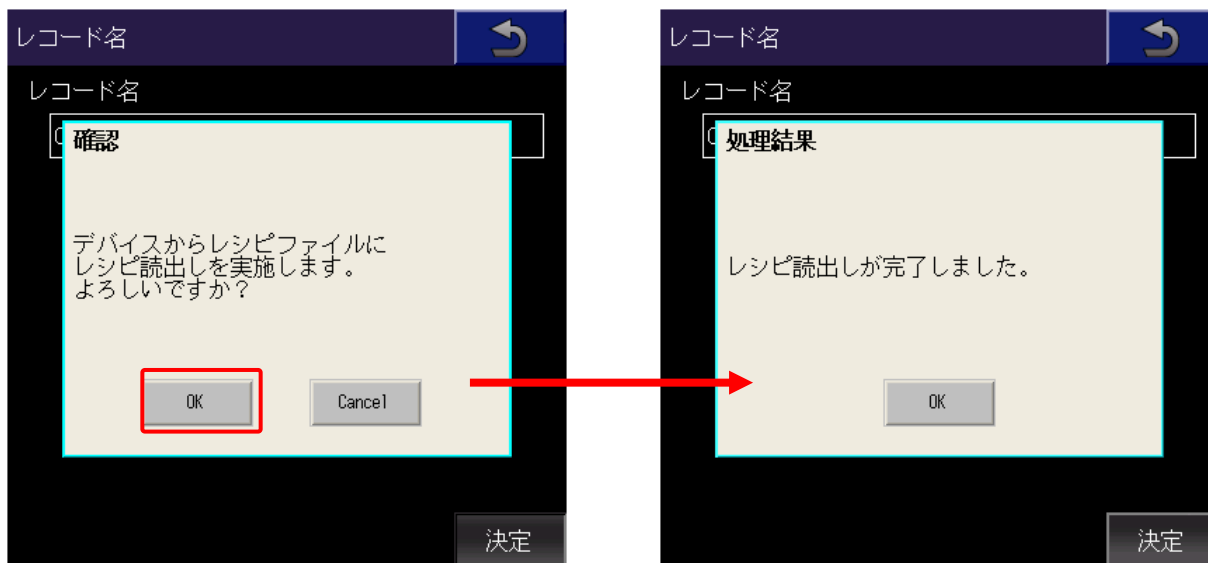
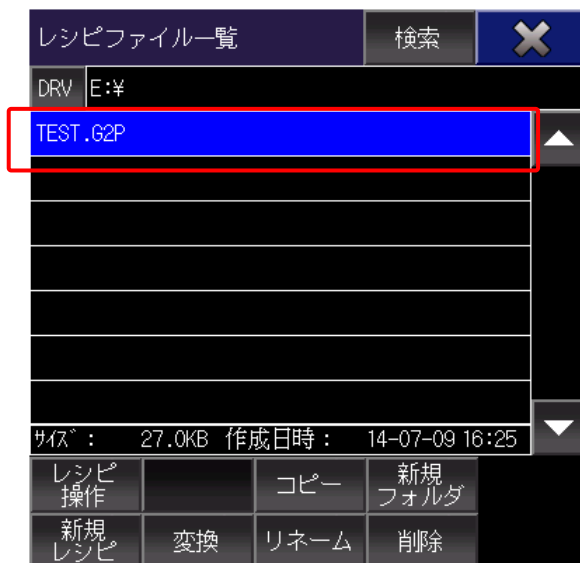
8.1 運転データやパラメータを USB メモリや SD カードへ保存する方法

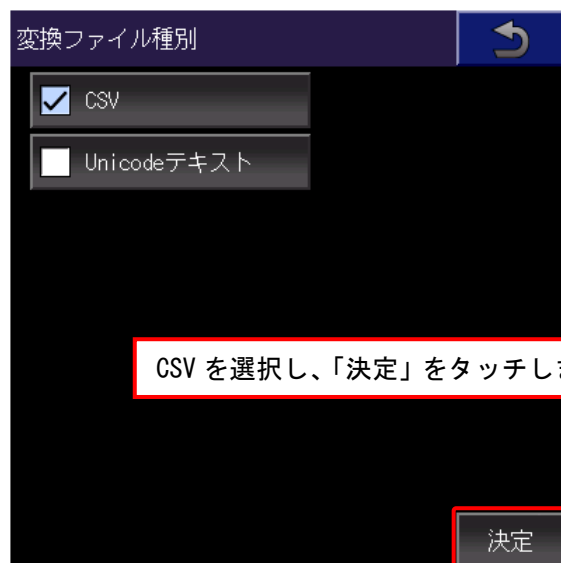
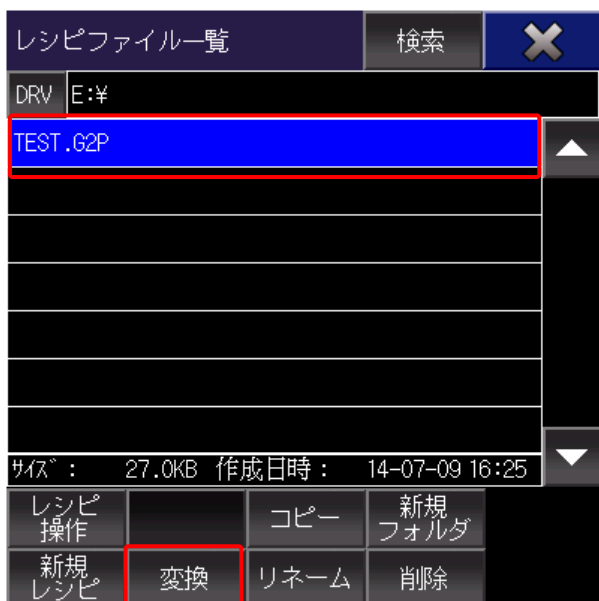
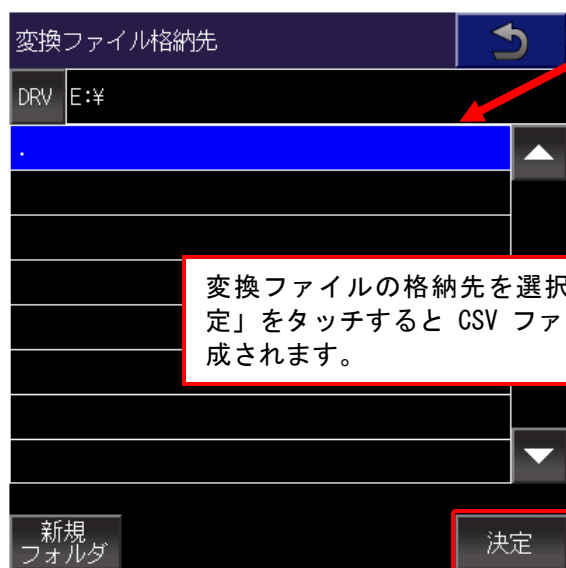
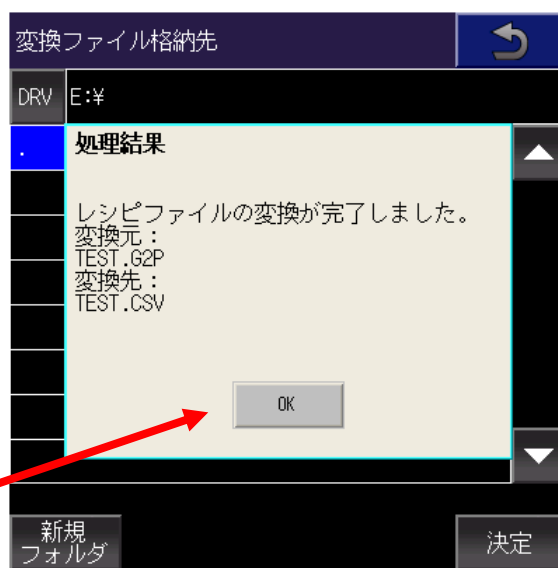
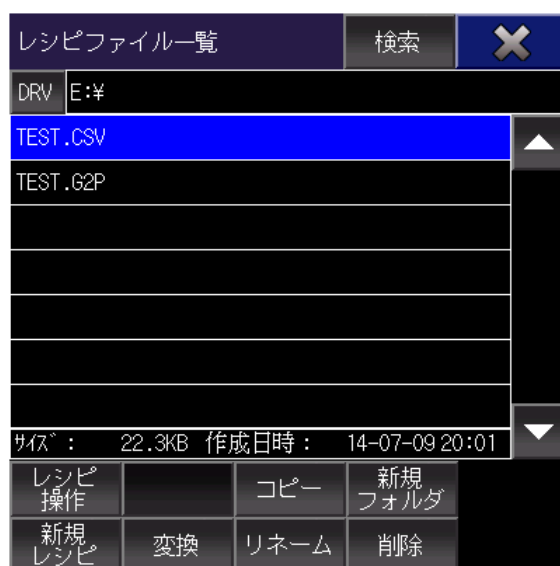
ここでは USB メモリにデータを保存する例を説明します。

(1) データ管理画面の「レシピファイル一覧」をタッチします。

(2) レシピファイル一覧のウィンドウが表示されます。ウィンドウの「DRV」をタッチしてください。

(3)　ドライブ選択が表示されます。E ドライブをタッチしてください。
その後、「決定」をタッチしてください。

(4)　新規レシピをタッチしてください。

(5)　新規レシピをタッチすると、下図のようにレシピ設定名(レシピ名称)一覧が表示されます。

レシピ名称の意味合いは、以下の通りです。

(6) USB メモリに保存したいレシピ名称(ドライバ号機番号ごとの運転データもしくはパラメータ)を選択します。
選択後、「ファイル生成」をタッチします。

(7)　レシピファイルが生成されると、下図にレコード名一覧が表示されます。
ここでは、「Operation Data」(運転データ)を選択します。
「Operation Data」(運転データ)を選択後、「デバイス->GOT」をタッチします。

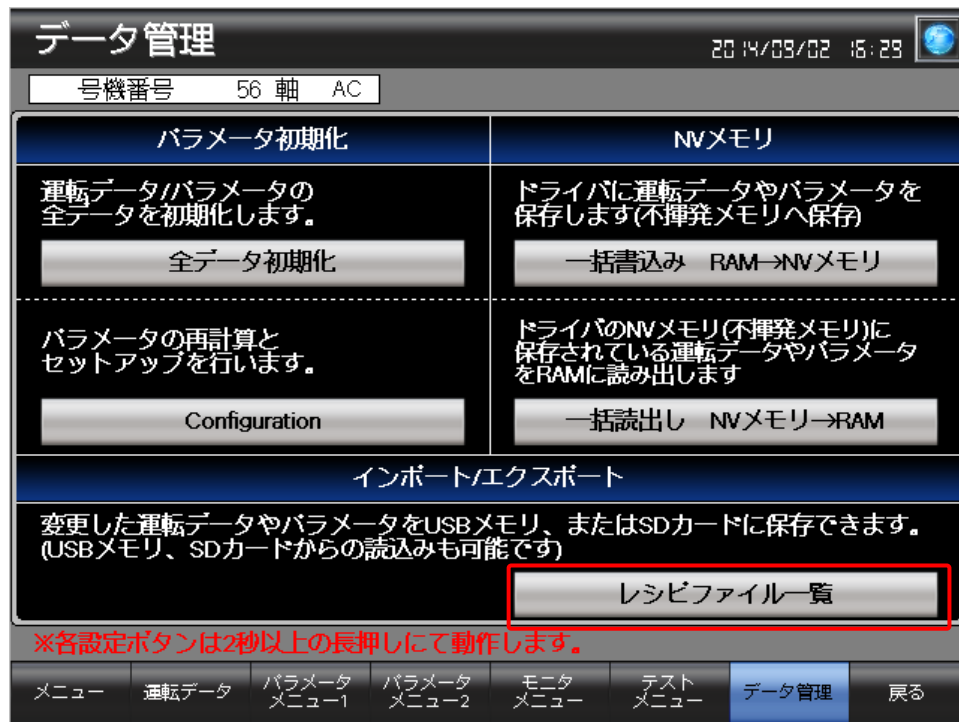
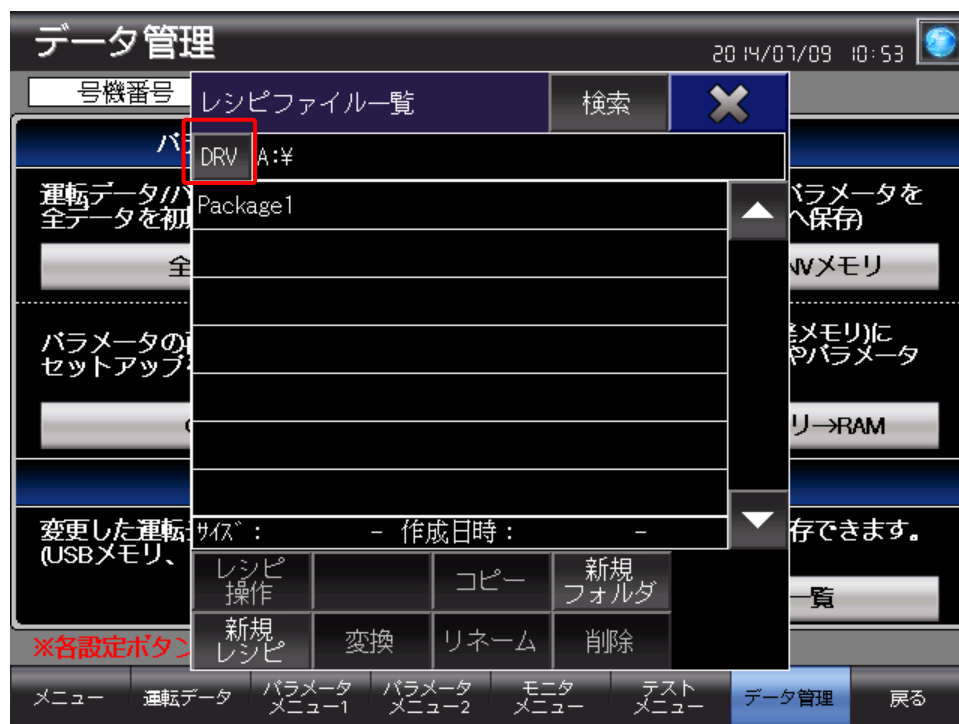
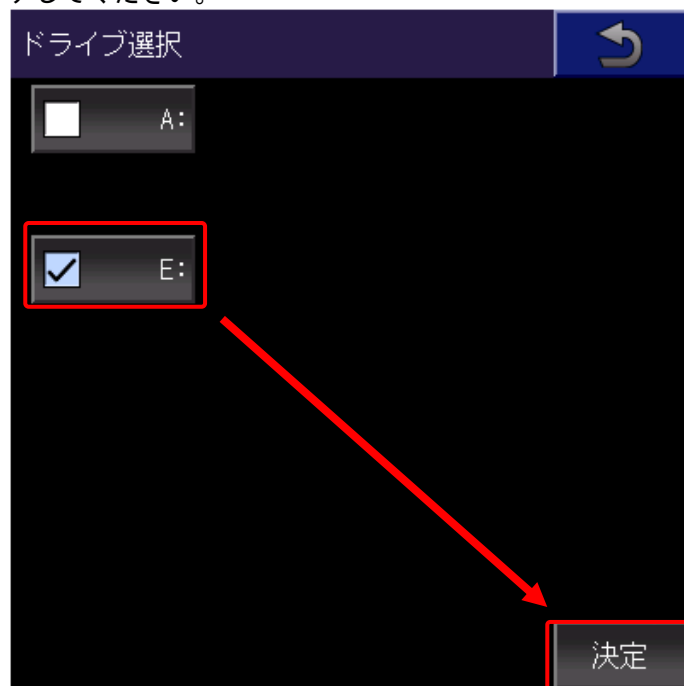
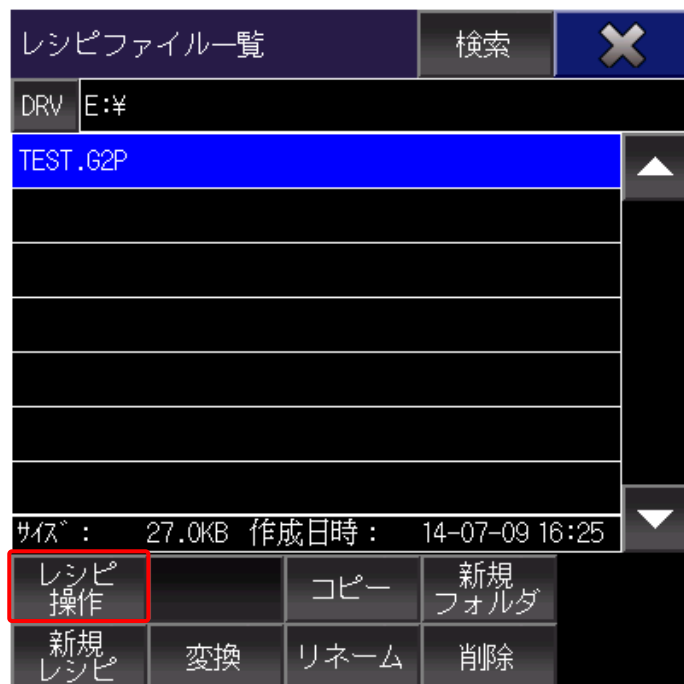
(8)　下図のように、レコード名設定画面が表示されます。
特に、レコード名を変更しない場合は、そのままで、「決定」をタッチします。

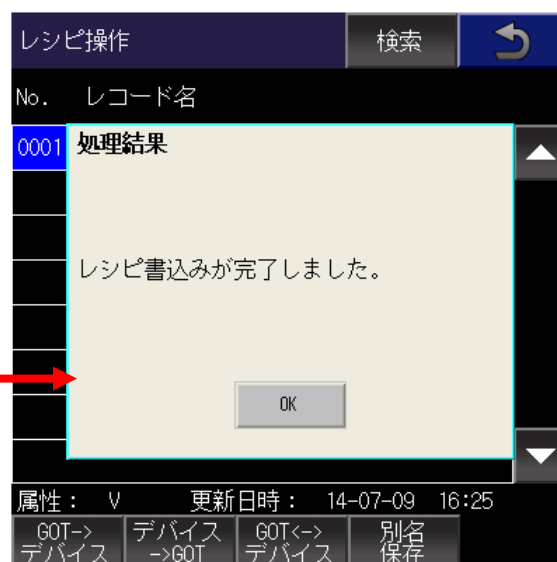
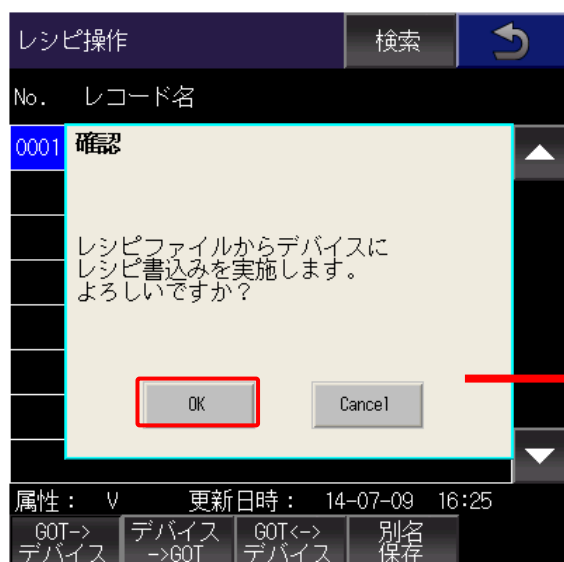
(9)　「OK」をタッチして、レシピ読出しを実行します。

(10)　レシピファイル一覧の画面まで戻ると、USB メモリに、ドライバから運転データを読み出したレシピファイルが生成されていることを確認できます。

*参考情報：ドライバ(AC 電源)のパラメータを保存したい場合は、(6)で「Driver Parameter AC」を選択後、(7)で「Parameter Data AC」を選択します。

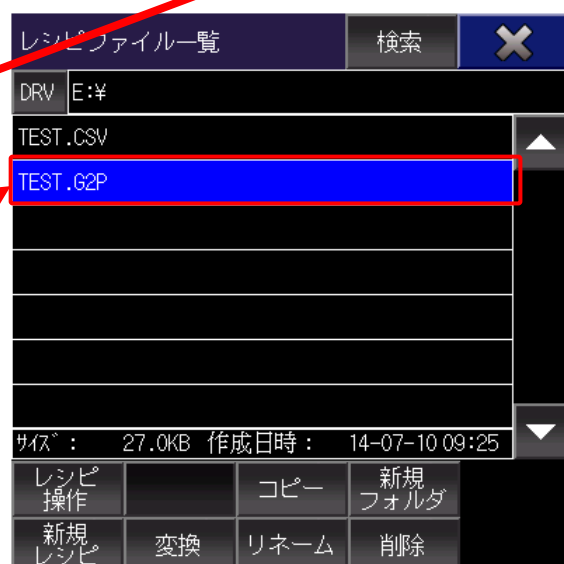
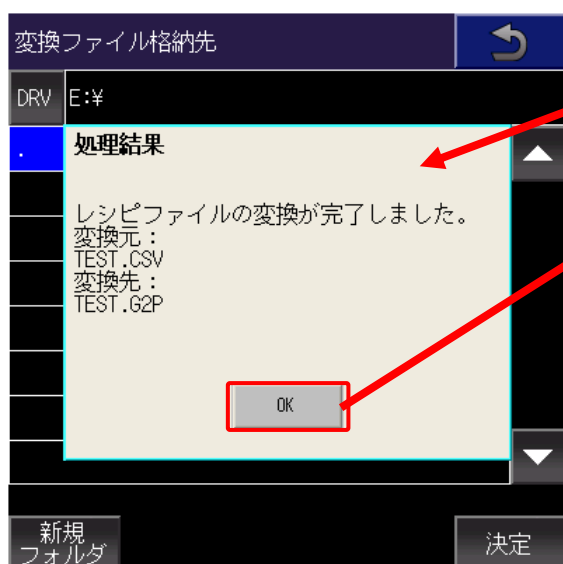
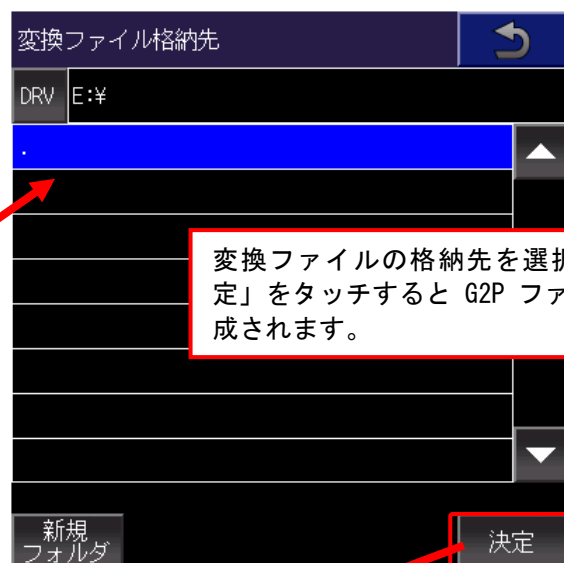
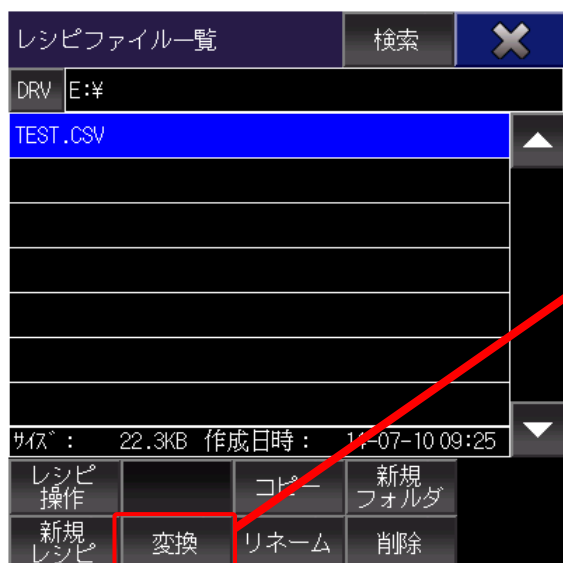
(11) 生成されたレシピファイルをパソコンで編集できるように CSV ファイル形式に変換します。
変換したいレシピファイルが選択されている状態で、「変換」をタッチします。

(12) 上記操作にて、下図のように、CSV ファイルが作成されます。

以上の操作にて、ドライバの運転データやパラメータを USB メモリに保存することができます。

8.2 運転データやパラメータを USB メモリや SD カードから読出す方法

ここでは USB メモリに保存されている運転データをドライバに書込む例を説明します。

(1)　データ管理画面の「レシピファイル一覧」をタッチします。

(2)　レシピファイル一覧のウィンドウが表示されます。
　　ウィンドウの「DRV」をタッチしてください。

(3)　ドライブ選択が表示されます。
　　Ｅドライブをタッチしてください。
　　その後、「決定」をタッチしてください。

(4)　書込むレシピファイルを選択し、「レシピ操作」をタッチしてください。

(5)　レコード名を選択し、「GOT->デバイス」をタッチします。

(6)　「OK」をタッチし、レシピ書込みを実施します。

以上の操作にて USB メモリに保存された運転データをドライバに書込みすることができます。

(7)　パソコンで編集した CSV ファイルをレシピ操作で使えるように G2P ファイルに変換します。
変換したい CSV ファイルを選択し、「変換」をタッチします。

以上の操作にて USB メモリに保存された CSV ファイルを G2P ファイルに変換することができます。

9. GOT 上での MODBUS アドレスの指定方法について

GOT(GT Designer3)上で使用するデバイスは,使用する MODBUS 機器のアドレスマップに応じて GT Designer3 上でのデバイス番号に置き換えて使用してください。MODBUS アドレスの詳細については、「GOT2000 シリーズ接続マニュアル（マイコン･MODBUS･周辺機器接続編)」を参照してください。

9.1 アドレスの置き換え方法

保持レジスタのアドレス「1234H」をモニタしたい場合、保持レジスタは GT Designer3 上ではアドレス「4*****」になります。GT Designer3 ではアドレス番号は 10 進数を使用しますので、「1234H」を 10 進数に変換して「04660」となります。また、GT Designer3 上のアドレス番号は、保持レジスタの場合「1」からとなりますので、アドレスに「+1」したアドレスとなります。よって保持レジスタのアドレス「1234H」は、GT Designer3 上では「404661」となります。

例：運転速度 No.0 の数値を表示する場合

運転速度 No.0 の上位の MODBUS レジスタアドレスは、10 進数で「1152」です。
*データ形式は、[符号なし BIN32]を設定しますので、運転速度の No.0 の上位アドレスを指定します。

実際に GOT の保持レジスタで設定するアドレスは、「1152+1」 = 1153
また、GOT の保持レジスタの指定アドレスは、「40000+01153」 = 401153 となります。